

第一章 論

1.1 研究動機與背景

標準的制定以及資訊技術的進步，使得負責解決問題的資訊系統。因為需要適應總體環境的變遷，才能使得資訊系統運作更有效果或效率。所以資訊應用的腳步必然需要緊緊跟隨資訊技術或是標準的發展。利用傳統的紙張作業、電話、傳真機以及無紙化的 EDI，皆是以往企業進行商務的資訊載具。但是網際網路串接全球各個角落之後，使得企業因為此網路的普及以及成本低，所以在上面從事商務行為，即所謂電子商務。但是傳統解決商務問題的資訊技術或標準，並不適用在網際網路的環境中。所以人們必須探討如何在新的環境中，制定新的標準及研發新的技術，以滿足新環境中的生存特質。這是本論文研究的主要動機。

利用網際網路作為傳輸媒介的電子商務，已經成為企業爭相進入的領域。大家皆相信，電子商務不再只是促銷公司的產品，它更是產品的銷售通路。以 Web 為應用平台的電子商務，一開始是由賣方自建網站來主導。然而，一個公開競爭而非少數賣方壟斷的市場，賣方的品牌知名度不足以吸引多數的顧客進來。因此，由多數的賣方和多數的買方所共同參與的電子交易市集，提供了買方及賣方更多的機會。這個新型態的組織將影響未來電子商務的發展甚鉅。本論文將研究如何建構一個應用在電子交易市集環境中的型錄搜尋代理人，在設計階段可以採行何種平台以支援它的實作。

從網際網路的歷史演進告訴世人，簡單及彈性是最重要的。任何開發技術如果違背這二個原則，將容易遭受失敗而消失匿跡。所以對開發一個網際網路的應用程式來說，掌握簡單及彈性的原則為第一要務。其次才為系統的效能，系統的效能隨著電腦硬體的快速發展，使得程式效能的差異性不大。而簡單可以讓終端

使用者以及系統開發人員，因為容易理解而有高接受度。而彈性的原則是要求用戶作最小的設定安裝以及原有架構的變動，使得不同的用戶端環境透過系統的代理機制完成系統的目的。

簡單的原則可以使用 XML 架構來完成，而彈性原則是本研究架構的特性。綜觀許多實際運作的電子交易市集，或是專家學者所提出的電子交易市集架構，即使功能強大但是卻未灌注太多的心力在用戶端的使者介面，以及市集參與者加入市集的彈性。在企業對企業的電子交易系統中，更是要求買賣雙方必須安裝一致可以互通的平台，才能讓電子交易市集完成電子商務的目的。如果一個公司必須耗時維護龐大的交易平台，或是因為安裝交易平台而必須更改原有的採購程序或是銷售程序。這對於那家公司來說，進入和退出市集的成本是相當高的。因此，必須設計一個彈性應用的架構，來提供電子交易市集解決方案。

並不是所有的產品市場皆可以採行電子交易市集架構，一個壟斷市場的供應商，吸收市場絕對多數的產品用戶，這種產品市場短期之內就不可能有成功的電子交易市集出現。電子交易市集要如何運作，才能讓參與者因為參與市集而取得優勢。這是企業競爭策略所關注的焦點，也是企業資訊管理的未來重點。只有企業具備完善的資訊管理平台，才能支援電子交易市集功能。資訊管理領域考量企業是否需要引進新的資訊技術以解決問題。以及決定引進新的資訊技術之後，企業要如何更變組織架構以調適企業文化上的變遷，以避免可能的衝突。所以本研究的主要動機，即在設計一個架構使得引進電子交易平台的企業，不會因為引進後變遷成本高，以及實際運作時造成與原有架構衝突。

1.2 研究目的

本論文主要目的在設計一個具高應用彈性、高效益、低導入成本的型錄搜尋代理人架構，並將之應用在電子交易市集的環境之中。如此的型錄搜尋代理人架構，可以促使應用它的電子交易市集，提供一個參與者彈性參與的環境。因為它沒有忽略賣方原有的銷售平台以及買方原有的採購平台。一個電子交易市集本身應該提供一個代理人機制，透過這個機制買賣雙方將只要向市集提出請求即可，內部的處理全交由市集的此機制自動完成。所以電子交易市集本身，可以視為軟體代理人系統的應用。何謂軟體代理人，將在下章作詳細介紹。本研究採用軟體代理人領域中的行動代理人，透過行動代理人來代替用戶身份完成派任的任務。在本研究架構中，行動代理人架構負責完成型錄代理人非同步回應模式。使用行動代理人架構的目的，從技術觀點來看，它提供更高效率的服務；從用戶觀點來看，它提供高品質的服務。

本論文主要的貢獻在即在「型錄搜尋代理」及「彈性參與」二部分。這是由以 Web 為基礎的行動代理人及三層式架構透過整合而完成，其中 XML 扮演訊息交換的角色。行動代理人並非一個運用普遍的資訊技術，然而它卻出現在多數的電子交易市集研究文獻中，可見它是一個極適用於電子交易市集的資訊技術。它之所以無法普遍應用在實際的電子交易市集之中，推論其原因應是它的安裝成本所至。行動代理人系統可以自主地完成使用者交託任務，足見其功能複雜性。由於其複雜性高違反了網際網路應用必須簡單才能成功的原則，所以本論文研究如何解決行動代理人複雜所引起的使用障礙問題。作法是借由本研究提出的整合架構，將原先全由行動代理人完成的架構，變成適合 XML 架構地方改成 XML 來取代，以降低行動代理人的複雜性。所以本研究才以行動代理人建構的電子交易市集架構為基礎出發，再研究 XML 可以幫助改良的地方。

1.3 論文章節概述

本論文第一章為緒論，表述為何要研究這個論文的動機，在動機中提出問題所在，而其後的研究目的提出解決方法的理念，以期建置一個架構來解決先前提出的問題。爾後則是本節，描述本論文的研究結構。

第二章的「電子交易市集應用環境分析」，是本論文所提架構的應用環境的分析。何謂電子交易市集，如何形成電子交易市集，以及電子交易市集發展的背景及歷史，其運作型態可以分成那幾種。針對這幾點的研究討論，以瞭解電子交易市集的特質，以及會面臨那些問題需要何種適合的架構來解決，本論文即在研究這個解決架構的內容。

第三章的「行動代理人架構」，提出行動代理人具備有軟體代理人以及分散式運算架構的特性，針對這二個特性進行討論。然後探討行動代理人的優點以及它適合應用的領域。再來探討我們可以利用什麼系統來開發行動代理人，以實作此架構。

第四章的「XML 架構」，探討 XML 用來描述資訊以及描述應用程式之間的訊息的方法。XML 架構適合用在應用程式整合的目的，尤其是 Web 平台的應用。在電子商務的應用上，XML 已經成為一個極重要的文件交換標準，有數個商務領域中已經建構適用於該產業的 XML 商務標準，對此將在本章作說明，並且解釋電子交易市集導入 XML 商務標準的重要性。

第五章的「型錄搜尋代理人架構需求與架構方案研究」，在探討型錄搜尋代理人架構的需求為何，建立一個基本的參考模式。然後利用此基本模式，列出各

類架構方案，供參考與研究以分析其優缺點，列出改進的目標。

第六章的「型錄搜尋代理人架構設計」，在研究如何將第三章所提的行動代理人架構及第四章的 XML 架構，這二個資訊技術應用在電子交易市集中型錄搜尋的應用領域中。在前一章中列出各類參考模型，在本章中設計一個整合各類架構優點的模型架構，它是一個整合行動代理人與三層式架構的模型，以 Web 服務為基礎，並以 XML 建立應用程式整合架構。然後將本章所設計的类型錄代理人架構，與各類由學者提出的架構，同樣應用在電子交易市集環境中的型錄管理，然後作一個詳細的分析比較。

第七章的「資訊產業應用與系統實作」，依據第六章所提的系統架構理念，將之具體化的實作，完成後評估其系統的價值何在。本章以個人電腦作為此電子交易市集系統的交易產品，舉例說明究個人電腦的產品市集，應用本型錄搜尋代理人架構後，其運作方式以及預期成效為何。

最後，針對本論文的分析架構的研究心得部分，以及系統架構設計和實作過程中，體驗到本研究對電子交易市集應用的價值部分。於第八章的「結論」作專章研究。從上面的章節概述可知本論文的研究分析架構，第二章談述電子交易市集的目的在清楚分析型錄搜尋代理人的應用環境;第三章的行動代理人架構及第四章的 XML 架構，為第五章和第六章建立學理基礎;第五章在研究架構的需求，並且提出各類文獻架構供設計時參考;第六章是本論文的架構設計核心，是第五章列示的架構之改良與整合;第七章則將第六章論述的架構實作在 PC 採購上，以證明其可實用性。

第二章 電子交易市集應用環境分析

2.1 電子交易市集精義

關於電子交易市集一詞，並沒有具公證性解釋，在此茲給予一個定義。「Electronic Marketplace」簡稱「e-Marketplace」，中文譯為「電子交易市集」或「虛擬商場」。本論文將一致採用「電子交易市集」一詞來表示。為清楚表達電子交易市集的涵義，茲將 e-marketplace 分解成”e”以及”Marketplace”二個字彙進行討論。隨著網際網路的興起，使得以”e”為前置字根的名詞快速擴增，比如 e-mail 代表電子郵件、e-business 代表電子企業。諸類名詞代表的意思是將傳統的作業進行「電子化」的再造工程。電子化是以網際網路為基礎(internet based)的同義詞，代表以網際網路架構為技術基礎，將作業流程自動化、數位化，以提升效率或節省成本。

“Marketplace”代表買方及賣方交易商品的場所，雖說“market”也可以代表交易場所之義，但是”market”所蘊涵的意義甚廣，而”marketplace”卻更能貼切表達交易場所之義。這是因為”marketplace”在”market”之後加上“place”字根，其解釋意義是場所、通路之義。”Marketplace”在此譯為交易市集，它代表的意義即是存在買方與賣方之間的虛擬通路，透過這個虛擬通路所提供的加值服務，買賣方雙方可以提升產品的流通效率及管制績效。交易市集是傳統既有的實體交易場所，雖說電子商務發展之初，強調顧客可以直接與供應商透過網站進行交易，不需要交易市集這類中間商存在。但是實際上，對產品品牌知名度低的供應商來說，即使網站可以直接提供交易的服務，但卻不易吸引顧客上網下訂單。將來即使電子商務的經營模式被商業社會所廣泛接納，但是存在於供應商與顧客之間的第三者，並不會消失而會因應時代的變遷而轉換成另一種型式存在。這類型式即

是以網際網路為基礎的虛擬交易市集，亦即所謂的電子交易市集。

因此電子交易市集，是將市集以網際網路加以虛擬重現，而目前雖然多數電子交易市集未提供傳統市集所有的機制。但是從網路商店的發展歷程，可知電子交易市集的未來趨向。早期的網路商店僅提供商品型錄介紹，到後來的網路商店才逐步導入交易機制，足見利用網際網路要取代傳統的組織，並非一蹴可成，而是循序漸進的。

由於商業環境的變遷，電子商務的經營型態隨之不斷改變，「電子交易市集」即是在這個變遷的產物。它在電子商務領域中扮演舉足輕重的角色，這點在第二節的「電子交易市集產生背景」中說明。電子交易市集依其經營產品類別、經營方式的不同，而孕育出不同的模式，這部分在第三節的「電子交易市集型態分類」中說明。在企業對企業電子商務中，有許多企業所支持成立的機構負責標準的建立，以求達到企業交易系統互通的目的。就專業研究單位的預測，電子交易市集是推動企業對企業電子商務的主要力量，且幾年內企業對企業電子商務的交易，有半數以上是透過電子交易市場來完成。到底電子交易市集有何對現今的商務環境有何貢獻，詳細內容在第四節的「電子交易市集的貢獻」作說明。電子交易市集的概念是歐美西方國家所建樹，即使是電子商務的模式運作在台灣這類亞洲國家，不見得可以適用，更何況是電子商務範疇中的電子交易市集。因此，在第五節探討的「我國企業電子型錄應用及電子交易市集發展概況」，是針對台灣企業環境導入電子交易市集的近況作論述。針對本章電子交易市集的論述，將在最後第六節作「小結」。

2.2 電子交易市集產生背景

企業建立一個提供商品目錄查詢的網站，將有助於企業的商品的促銷，但對企業的營運績效提升效果不大。但是，企業如能利用現今的資訊網路科技，將電子商務經營視為企業的通路經營模式，將能發揮電子商務真正的核心價值。傳統企業必須建立配銷通路，將商品交託至自營或授權代理的配銷商，然後配銷商再將此商品銷售至顧客。如今，採用電子商務經營模式的企業，可以建立一個虛擬的交易場所，使得商品可以透過此場所直接與顧客接觸。企業可以因此而接洽到更廣大的顧客，並促使商品供應鍊上的企業必須進入這個領域中。依據 Mark Norris 於「eBusiness Essentials: Technology and Network Requirements for the Electronic Marketplace」一書中的定義[19]。這個虛擬的交易場所，稱為電子交易市場(electronic market)。之所以謂之虛擬是因為它沒有具體的店面，交易雙方透過網路傳輸及電腦螢光幕顯象，即完成交易的程序。取而代之的電子交易市場的商場店面是一頁頁的網頁，商場的貨架等於是網頁呈現的目錄，所有商品皆置放及展示在上面。而商場所需的倉儲中心則是存放商品資料庫的伺服器。以資訊流來取代物流的呈現，直至送貨到府，顧客才真實接觸實質貨品。

綜合「eBusiness Essentials」一書所提的電子交易市場的主導者分析，以及 CommerceNet 的「The Development of Online Marketplaces」[28]研究報告內容。本文將電子交易市集(electronic marketplace)產生的背景作以下描述。

在電子交易市場中，因為市場的主導者不同，而產生不同的市場型態。如果說市場是由少數具有品牌魅力的供應商，來吸引多數的買方進入此市場，則它被稱為“賣方主導”的市場。反之，如果說市場是由少數大型的顧客吸引大量的供應商加入此市場，則它被稱為“買方主導”的市場。但是，如果說買賣雙方皆為

數眾多，且沒有明顯固定的賣方或買方得以主宰市場，一切由市場機制來決定商品價格，則此市場稱為“開放”的市場，易由第三者來經營此市場。利用第三者建立的電子交易市場的經營模式，即稱為「電子交易市集」。

賣方主導的市場因為少數具規模的賣方供應商，恃其品牌知名度吸引多數的買方顧客進入此市場，所以供應商會各自建構伺服器作為電子交易平台，稱為「賣方產品型錄」(Sell-side Catalogs)的模式。顧客在選擇條件有限的環境下會選擇這些供應商來進行採購。主導此類市場的供應商，會期待在交易過程中，創造和維繫市場和價值，所以會提供詳盡的附加服務，如在網頁中提供顧客所訂的商品的組裝進度，或是確認送貨的狀態。

買方主導的市場是由少數具規模的買方顧客，恃其強大的購買力吸引多數的賣方供應商爭相進入市場，這種型態的市場在企業對企業的電子商務中尤其多見。在企業對企業電子商務的市場中，企業顧客對上游工廠的採購產品可將之畫分成二種類別：第一種稱為關鍵產品，亦稱為產品零組件，是構成製成品的實質元素。第二種稱為非關鍵性產品，亦稱為非生產性物料(MRO, Maintenance, Repair, Operations)，是輔助生產的相關設備，比如辦公設備、機械潤滑油等。由於第一種影響產品製造甚鉅，許多顧客對購買之商品的可靠度有相當的要求，因而目前利用電子商務模式推展此類產品市場較為不利。但是第二種由於選擇性高加上無第一種的問題，所以形成多數的供應商可供買方選擇，因而此類產品的交易市場易形成買方主導。買方主導的市場中，企業利用「買方採購前端」(Buy-side Procurement Front-Ends)的軟體，來規劃其採購貨品的交易對象。

在買賣雙方皆無能力主導市場時，此市場即稱為開放市場。此類市場是由第三者來經營，這個第三者即所謂「電子交易市集」經營者。開放市場可以分為私有式開放市場和公眾式開放市場，二者皆由第三者或是由多家廠商共組的企業來

作為電子交易市集的促成者、匯聚者。私有式開放市場提供一個封閉的買方虛擬社群，此社群稱為「採購交易市集」(Procurement Marketplaces)，亦稱為私有式電子交易市集(Private Marketplace)。交易的標的物多為非生產性物料，前述所論的買方主導的市場，很容易由買方匯聚形成此私有式的開放市場。透過這個私有式的開放市場，買方企業在該類產品上產生極強大的買方市場力量，因而吸引供應商紛紛加入此市集，以提供最佳價位的商品。在非生產性設備產業中，由 Ariba 及 CommerceOne 這二家主導電子交易市集發展的領導廠商，將改變企業對企業關於非生產性設備採購的生態。而透過他們所研發的應用系統，企業內部對於非生產性設備的採購將會透明化，顧客提出設備需求之後將透過市場，找出能提供此顧客預期價位的供應商。由美國三大汽車廠福特、通用以及克萊斯勒集資備成立的「Covisint」電子交易市集，將經營一個由強大買方主導的汽車零組件的私有式開放市場。此類市場組成的目的在聚數個顧客的集體議價的能力。對於供應商來說必須面臨削價競爭才能獲取顧客訂單，所以此類市集無法扮演公正第三者角色。

另一方面，公眾式的開放市場不同於私有式的型態，因為前者是由大型的買方聚合一個電子交易市集，來吸引多數的供應商加入其經營的市場。而公眾式的開放市場，超脫從事產品的買方或賣方角色的企業，所成立的電子交易市集來經營公眾式開放市場，此類市集稱作公眾式電子交易市集(Public Marketplace)。私有式的開放市場對會員參與的限制條件高，產生有意參與廠商的進入障礙。而公眾式的開放市場提供了商品入口網站的功能，讓相關的商品聚合於此，買方可以有存取線上供應商的機會。此類電子交易市集可以依靠軟體代理人的資訊技術，促進電子交易市集的資訊平台與參與廠商的平台進行系統整合。顧客可以自訂採購清單，不需要有該產品的領域知識，即能透過軟體代理人的協助，來決定顧客應該買什麼商品以及向誰買。在公眾式開放市場中，產品的買賣雙方透過公正的電子交易市集進行商品交易。所以易吸引全球的企業參與，對有意擴充產品市

場廣度的企業來說有很高的助益，交易安全以及企業徵信將是該市集需要著眼重心。強調眾多買方與賣方參與的公眾式電子交易市集，被大多數人認定為真正的電子交易市集。所以狹義的電子交易市集，即為公眾式電子交易市集。

根據上述 Norris[19]對電子交易市集產生背景的分析，結合 Terry[28]對相同議題的論述。筆者將電子商務發展歷程作為橫軸，市場生態及電子交易市場出現的經營型態作為縱軸，將電子交易市集的產生背景歸納出一個概觀的矩陣，以圖 2.1 加以表述。

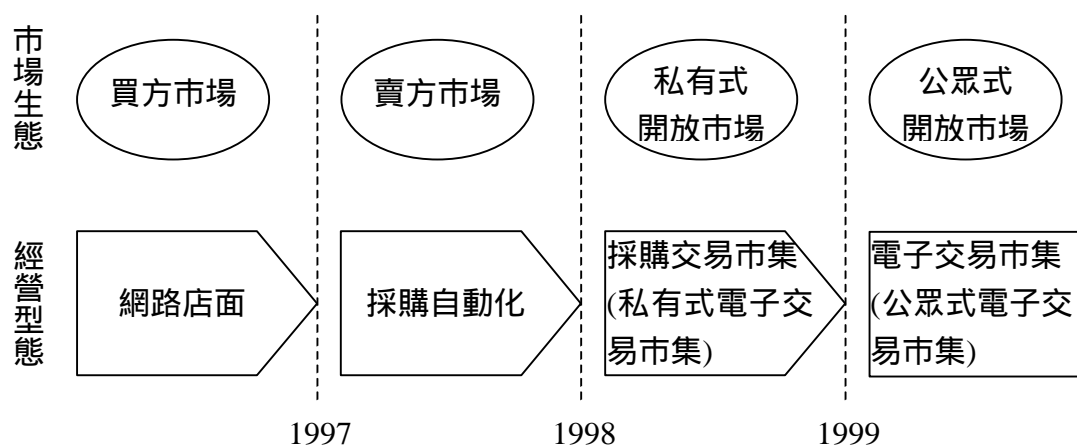


圖 2.1 市場生態與電子交易市場經營型態發展的歷程

一個能夠被大眾廣為接受的電子交易市集，應該經營公眾式的開放市場。因為電子交易市集應該要保持超然客觀中立的立場來經營其市場，才能讓它可以在交易增值服務上扮演更重要的公正第三者角色。實際上，電子交易市集不僅可以提供交易仲介，它可以更積極地提供其他增值服務[35]，如圖 2.2 所示。唯由顧客提出招標公告易造成供應商為爭取訂單而削價競爭，因而降低供應商參與電子交易市集的意願。所以本論文將在後述提出的型錄搜尋代理人架構中，採用公眾式電子交易市集模式來提升供應鍊中廠商間的存貨週轉率，並為顧客尋找最佳化價格的產品。為考量供應商參與意願，所以本市集不建議完整採用圖 2.2 的架構。此市集將只提供交易配對功能的型錄資訊，由供應商提供線上交易功能。

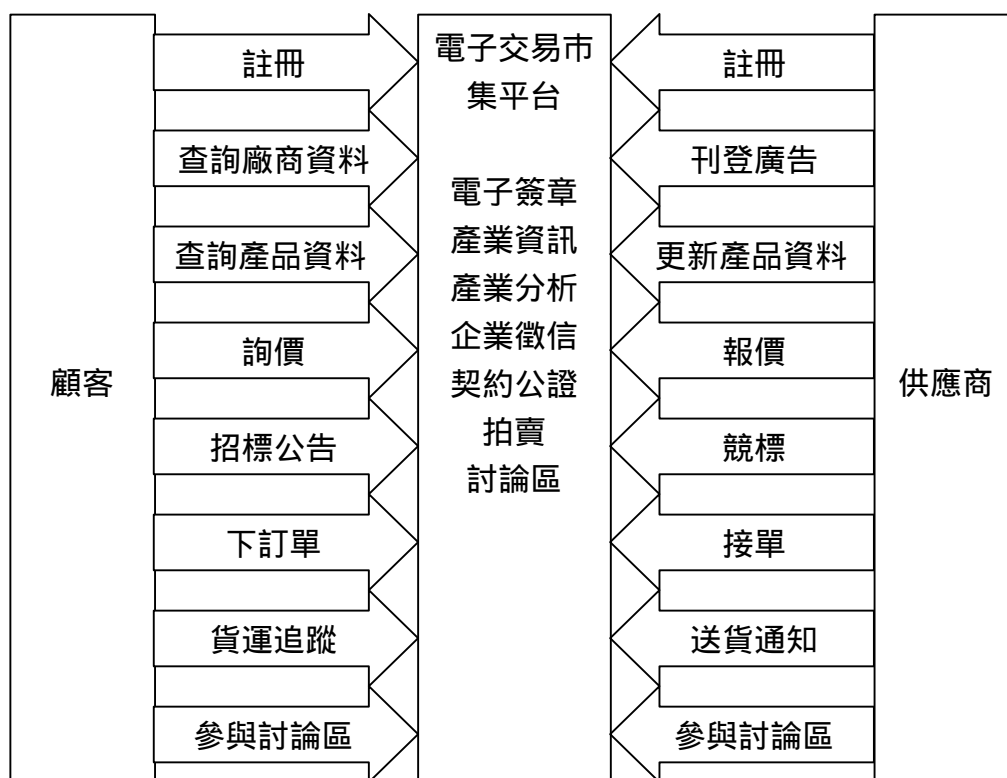


圖 2.2 電子交易市集架構[35]

電子商務的商業模式中，最具規模的是「企業對企業」以及「企業對消費者」二種模式。前者擁有量大客源穩定的特質，後者具有量小客源變化大的特質。根據美國 Forrester 市場分析公司的統計，未來五年之內前者的市場規模將數十倍於後者，可想見企業對企業的電子商務將是未來從事電子商務資訊業者的主要著眼之處。從企業對企業的電子交易市場來看，最早開始追溯到 1995 年。這年，從事此類市場的經營者之一的 Industry.Net，因為早期技術環境的缺乏加上其局限在需要付費才能加入的封閉市場，終告無疾而終。如今網際網路已經被許多企業所接受採納，企業對電子商務的接受度及技術成熟度都比過往幾年進步，因此發展電子交易市集有更佳的成功優勢。在過去企業對企業的商務往來，有 EDI 標準可以達到商業文件的交換目的，如今雖說各界認為傳統 EDI 即將走入歷史，但是其實各界發展電子交易市集的架構，卻有不少是依循 EDI 訂定的標準加以改良，所以 EDI 將會改變其型態而非正式的消失。傳統的 EDI 是在企業間專屬

網路上運作，而改良 EDI 的實作架構則遷移至網際網路上。而後在 1995 年開始萌芽的賣方主導的市場，以及 1998 年開始出現的買方主導市場，逐步取代 EDI 在企業對企業商務的地位。如今企業對企業的電子商務，將逐漸透過電子交易市集來進行。其成長速度比上述各類型態皆高出許多。根據 Forrester Research 調查，電子交易市集是將來企業對企業電子商務成長的主要動力，未來五年內將會有 53% 的企業對企業電子商務是透過電子交易市集。蔡桂芳與萬洪濤所著的「B2B 虛擬商場完全經營手冊」中[35]，將企業對企業電子商務的技術發展歷程，分為四個階段，以圖 2.3 表示。首先發展的 EDI 將買方及賣方透過增值網路以一對一的方法連結起來，進行商業資訊交換。接下來由賣方經營網路店面，透過網際網路與買方建立一對多的關係，提升其交易效率及接觸潛在客戶。再來的由買方經營的採購自動化介面，透過網際網路與賣方建立一對多的關係，提供其採購的品質與效率。最後出現的電子交易市集，提供買方與賣方共同參與的虛擬社群，使買方與賣方可以建立多對多的關係，建立買賣方雙贏的合作模式。這類電子交商技術買賣關係以圖 2.4 表示。

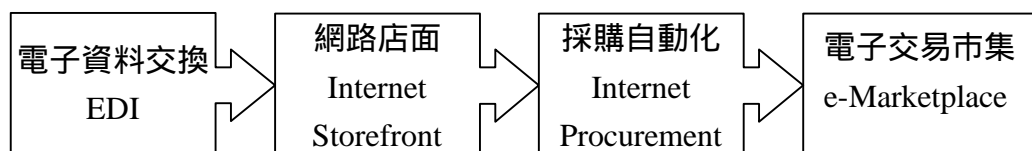


圖 2.3 企業對企業電子商務技術發展歷程[35]

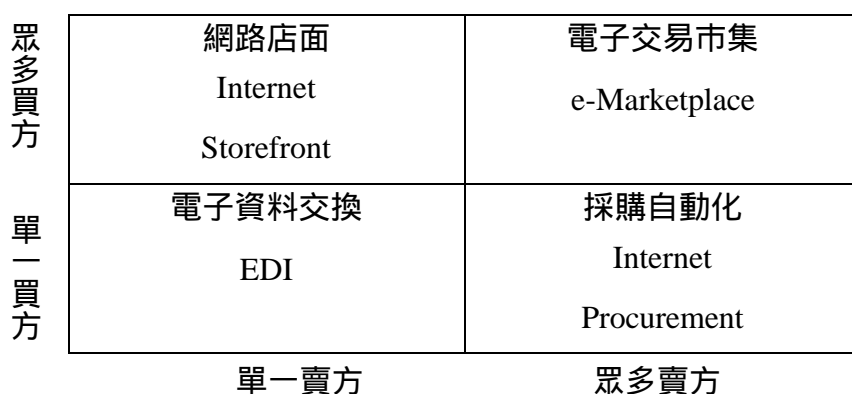


圖 2.4 各類技術與買賣雙方關係[35]

2.3 電子交易市集型態分類

簡言之，電子交易市集是一個聚集產品的買方及賣方的公開網路交易環境，而目前各類型的電子交易市集逐漸成形，但不同型態的電子交易市集，有一個永存的定律，此定律是人類希望可以更容易且更快速地探詢滿足本身需求的產品。依據 1999 年 Terry 在 CommerceNet 的研究報告指出，目前電子交易市集可以扼要地分為四種主要型態[28]：垂直式交易性市集、垂直式非交易性市集、水平式交易性市集、水平式非交易性市集。依商品種類區分，將垂直式 - 水平式市集當作縱座標；依商業模式(Business Model)區分，將交易性 - 非交易性市集當作橫座標，可以歸納出目前的電子交易市集矩陣，此矩陣如圖 2.5 所示。茲將四種型態的市集詳述其內容及特徵。

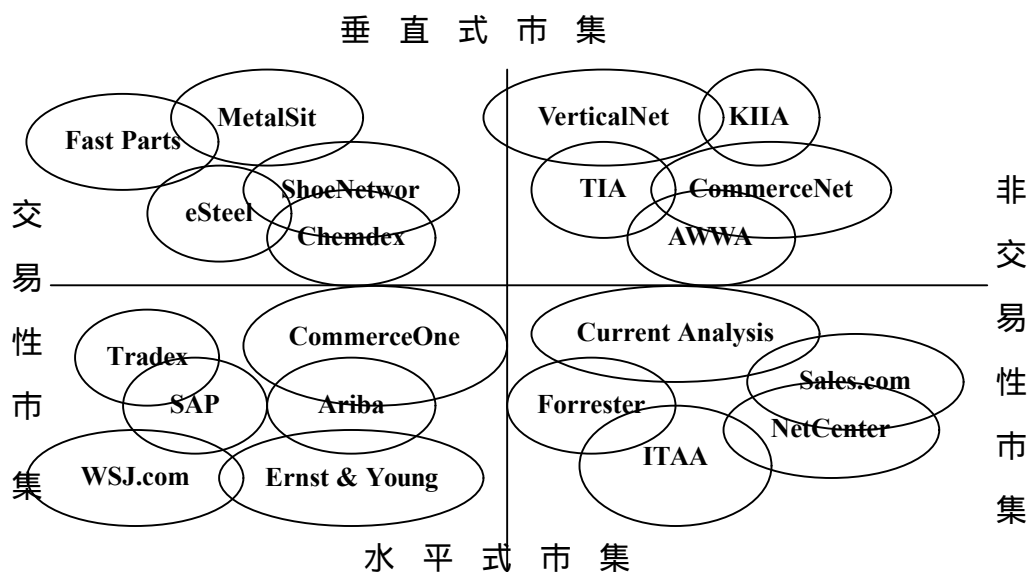


圖 2.5 Terry 的電子交易市集分類矩陣[28]

1. 垂直式交易性市集

垂直式交易性市集可以用在許多地理位置或是各個部門，是供應鍊上的企業因為供需關係而形成的市集，對於企業主管來說是相當誘人的商業策略途徑。此類市集在特定的產業中提供專業交易服務，所以市集的管理者必須對此產業相當熟悉。每個垂直式市集都有它固定的商業流程，而推動此類市集的管理者必須將此流程進行地更有效率，而並非變更它們。從事此類市集的廠商，其收入來源在入會費用及交易抽成。此類市集的領導廠商包含：從事生物科技產品的 Chemdex、鞋類生意的 Shoe Network、特製鋼鐵的 e-Steel、二手鋼鐵的 Metal Site 以及從事電子零件的 FastParts 等。

2. 垂直式非交易性市集

此類市集與前者的相同點在於它聚集特定產業的買賣雙方進入此市集，但最大的不同點在於此類市集本身不直接提供交易服務，它們是特定的單一或多個產業的入口網站，此市集是扮演凝聚者的角色，透過此市集的仲介促使買賣雙方得以完成交易。從事此類市集廠商的收入來源在諮詢費用及廣告收入。此類市集的領導廠商包括：從事多種產業的 Vertical Net、袋鼠生意的 KIIA、旅遊業的 TIA、電子商務的 CommerceNet、科技產業的 Red Herring 等。

3. 水平式交易性市集

水平式交易性市集可以稱為入口網站或是商務服務平台，水平式市集性市集的交易標的是間接性或稱非生產性的產品或服務，這些產品或服務是適用於所有的垂直產業或是單一企業，而服務的範圍更可以擴張到人力資源、後勤管理及支票付款等。由於這些產品沒有指定用在特定的產業，所以它可以服務的廠商就比垂直市集還要更廣。而企業透過水平式交易性市集來採購，預期可以省下 5-15% 的產品成本，70% 的人工處理成本及 50-70% 的採購週期時間。在電子商務被全

人類所廣為信賴與接受之前，相信水平式市集是比垂直式市集更容易被接受的。而水平式交易性市集由於在網站上提供交易功能，所以它的收入來源有部分是對廠商的交易抽成費用。另一方面有些水平式交易性市集提供用戶採購的套裝軟體，而這些軟體的授權費用成為其主要收入來源之一。目前在這類市集中的領導廠商有：Ariba、Tradex、CommerceOne 提供用戶採購自動化軟體，而 SAP、Wall Street Journal、Ernst & Young 亦是這類市集的代表廠商，但並未提供此軟體，其中傳統企業資源規劃軟體的領導廠商 SAP，將與 CommerceOne 合作以順利進入電子交易市集領域。另外，雖然 Ariba 和 CommerceOne 提供採購自動化軟體，但參與 Ariba 市集的廠商必須得到該軟體授權，而相對的 CommerceOne 則不強制參與廠商使用該軟體，CommerceOne 是以收取交易仲介費來經營。

4. 水平式非交易性市集

不同於垂直式非交易性市集的性質，此類市集是跨產業的資訊內容及社群服務的提供者，而相同點在於它們主要不是提供線上交易的市集。此類市集形成企業對企業電子商務社群，而此社群的形成是透過內容、資訊、意見、企業規劃及生產規劃等的凝聚。研究單位、線上新聞機構、商業協會等都可以被認定是此類市集。它們主要的收入來源在諮詢費用及廣告收入，目前這些機構包含有：從事科技調查分析的 Current Analysis、資訊業蒐集的 NetCenter、科技調查分析的 Forrester Research、科技協會的 ITAA、銷售支援的 Sales.com 以及財務報告支援的 WSJ.com。在國內，由經濟部技術處主導的 IT IS 計畫，整合國內參與科技專案之財團法人研究單位(包含資策會及工研院等)，提供各領域的專業市場情報，以供入會的廠商透過其提供的資訊，掌握市場動向。目前各類產業資訊已經蒐集相當齊全。

上述 Terry 定義的電子交易市集分類，依據商品種類以及商業模式作為區分標準。另一方面，在蔡桂芳與萬洪濤所著的「B2B 虛擬商場完全經營手冊」中[35]，

亦依據商品種類以及商業模式作為電子交易市集的分類標準，其中商品種類亦將電子交易市集區分為水平及垂直式。在商業模式分類標準上，將電子交易市集分為四種模式：型錄服務模式、效率提升模式、交易所模式、拍賣模式。依據商品複雜度及價格機動性的程度上的不同，可以區隔成四種利基市場。這四種不同的商業模式分別可以運作在各別的利基市集中，以圖 2.6 加以表示。這四個模式的運作方式，分述如下：

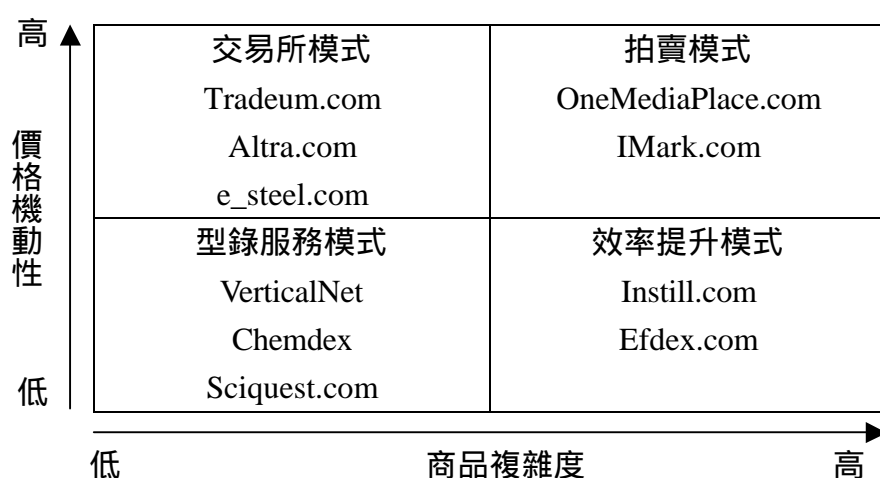


圖 2.6 各類技術與買賣雙方關係[35]

1. 型錄服務模式(Catalog Model)

利用網路提供商品型錄的服務，是早期電子商務的經營模式。而電子交易市集提供的型錄服務模式，是將供應商提供的產品透過電子交易市集的網站呈現，供顧客上網查詢。這類市集匯集供應商的廠商及產品資料內容，扮演商品型錄入口網站的角色，本身不提供交易機制。僅提供型錄服務不與企業爭利，是採用此模式經營的電子交易市集的利基市場。猶如上述的非交易性市集，提供各類產業商品型錄服務的 VerticalNet，是最具代表性的此類電子交易市集。

2. 效率提升模式(Efficient Commerce Model)

採用此模式的電子交易市集，以改善顧客與供應商的既存的商業關係來創造其價值。對於已經與上下游廠商有長久穩定交易關係的企業，以及標準化高的產品極適合。對於許多企業來說，非生產性設備採購願意透過電子交易市集來降低其採購成本，但是對關鍵零組件的採購則維持與可以信賴的企業合作。這類市集將會容易被已經有穩固交易關係的貿易夥伴所接納，並且替供應商尋找新的買主。比如，從事食品服務產業的 Instill 電子交易市集，制定標準化的產品型錄，替提供食品的供應商找到大型的連鎖旅館顧客。

3. 交易所模式(Exchange Model)

採用交易所模式的電子交易市集，透過替市場的需求面以及供應面進行配對來創造價值。此模式猶如證券交易所的模式，提供顧客出價及供應商叫價的配對技術。對於產品價格變動大的產業，電子交易市集可以採用此種模式來經營。從產品的角度來說，提供產品的供應商不只一家，所以此模式的電子交易市集，提供多個供應商及顧客來針對特定產品進行議價。產品本身具備標準化，再經過市集平台的產品資訊標準制定(如授予產品編號)，交易所模式的市集即能進行運作。從事鋼鐵產業的 e-Steel，以及從事紙張產業的 PaperExchange，為此種模式的代表性電子交易市集。

4. 拍賣模式(Auction Model)

採用拍賣模式的電子交易市集，利用替顧客及供應商進行配對來創造價值。同樣是採用交易配對模式的交易所模式，與本模式不大相同。從產品的角度來

看，拍賣模式所提供特定產品的供應商是特定一家，不像是上述的交易所模式是由多家供應商一同叫價。因此拍賣模式是一家供應商對多個顧客進行議價，而交易所模式是由多家供應商對多個顧客來進行議價，此其最大不同點。對於沒有產業標準的產品以及二手產品的產業，電子交易市集極適合採用此種模式來經營。提供二手機器的 iMark，以及提供網路庫存廣告服務的 OneMediaPlace，是採用本模式經營而具代表性電子交易市集。

2.4 電子交易市集的貢獻

2.4.1 對供應商及顧客的貢獻

電子交易市集的出現，是新一波電子商務的趨勢，尤其是其中的企業對企業的經營模式。企業賣方獨立建設的市集，將逐漸面臨挑戰，而聚集買賣雙方在一個作業平台上進行商務是企業的正确策略考量方案。Shim, Pendyala, Sundaram 等人在”Business-to-business e-commerce frameworks”論文中指出，電子交易市集帶給買賣雙方的實質利益，列示如下[26]。

- 顧客的實質獲得利益
 - (1) 與潛在供應商有更多接觸機會
 - (2) 採購部門可以專注在購買策略而非煩雜紙上作業
 - (3) 透過自動化的程式完成供應合約
 - (4) 得到分析採購方式的能力以提供企業洞察力
 - (5) 採購記錄的追蹤
 - (6) 降低人為造成的錯誤

- (7) 降低存貨及運送成本
- (8) 降低非合約式的購買
- (9) 提升資訊流以提供較佳的企業決策

- 供應商的實質獲得利益

- (1) 與潛在顧客有更多接觸機會
- (2) 開啟新的產品通路
- (3) 提升客戶服務的品質
- (4) 提升生產規劃能力
- (5) 降低顧客只購買單一價格的偏好
- (6) 提升運籌配送的資訊流

其餘第(7)至(11)項與前述的顧客的實質獲得利益中的第(5)至(9)項相同。

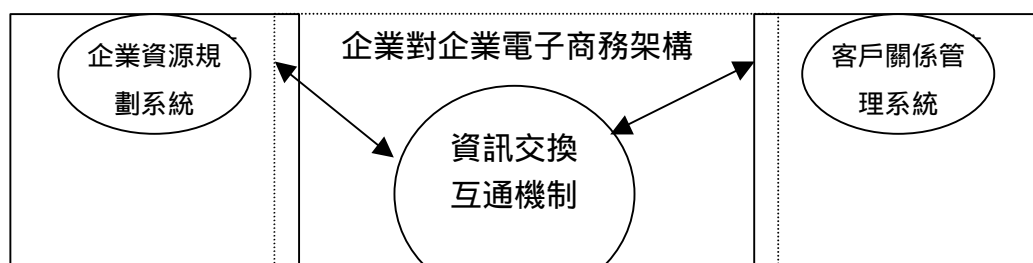
電子交易市集可以將顧客及供應商原先所需支付的固定支出項目的費用降低，使得企業營運更有績效。而透明化的價格、運送排程、品牌聲譽在此市集中更易被發現。便宜的價格配上輕易取得的資訊將可以提升顧客在商業電子化世界中的競爭力，這是不可避免的趨勢。

2.4.2 對企業資訊系統整合的貢獻

商務可以被視為許多相關程序的集合，而電子商務的本質及目的是在將原先的程序數位化及自動化，使得整體的商務可以更有效率。電子交易市集提供了電子商務可以運作的平台，而企業對企業的電子商務透過此平台更能將原先的企業內部應用系統進行資訊流上的整合，使得所有企業的內部系統不再只能為單一目

的而執行作業。這些企業內部的應用系統包含有企業資源規劃(ERP)、自動銷售(SFA)、客戶關係管理(CRM)、供應鍊管理(SCM)等系統[26]。發展企業對企業的電子商務必須著重在如何將這些原先的企業資訊系統進行整合,才能彌補這些系統交易上的一個缺口(如圖 2.7)。

電子交易市集的建構是企業的策略政策,搭配企業的資訊科技及資訊管理支援,將可以為企業帶來三個主要優勢:成本降低、效率提升、尋找潛在客戶。在技術上,欲完成企業對企對的電子交易市集,最重要的是制定訊息交換的規範。在考慮眾多廠商支援、易於實作開發、富於彈性及擴充性大等原則下,可延展性標示語言(XML)已經在眾多的交易市集中被引進、推廣,能夠整合各類資訊系統是 XML 最大的利器。比起傳統利用 CORBA 來達成整合的目的,以 XML 為基礎實作電子交易市集的資訊系統比用 CORBA 有更多的好處,這部分將在第四章進行說明,而可見的事實是原先 CommerceNet 使用 CORBA 來實作電子交易市集,但是已經在前幾年改成用 XML。另外,電子交易市集多是使用三層式分散式運算架構來實作,不論其訊息標準是否使用 XML。三層式架構有它的優點存在,它也是目前最被普遍使用的方式。但是它在績效上可以透過行動代理人的架構來進行改善,行動代理人的架構將在第三章說明,亦在此章節中說明如何運用行動代理人系統在設計電子交易市集架構。



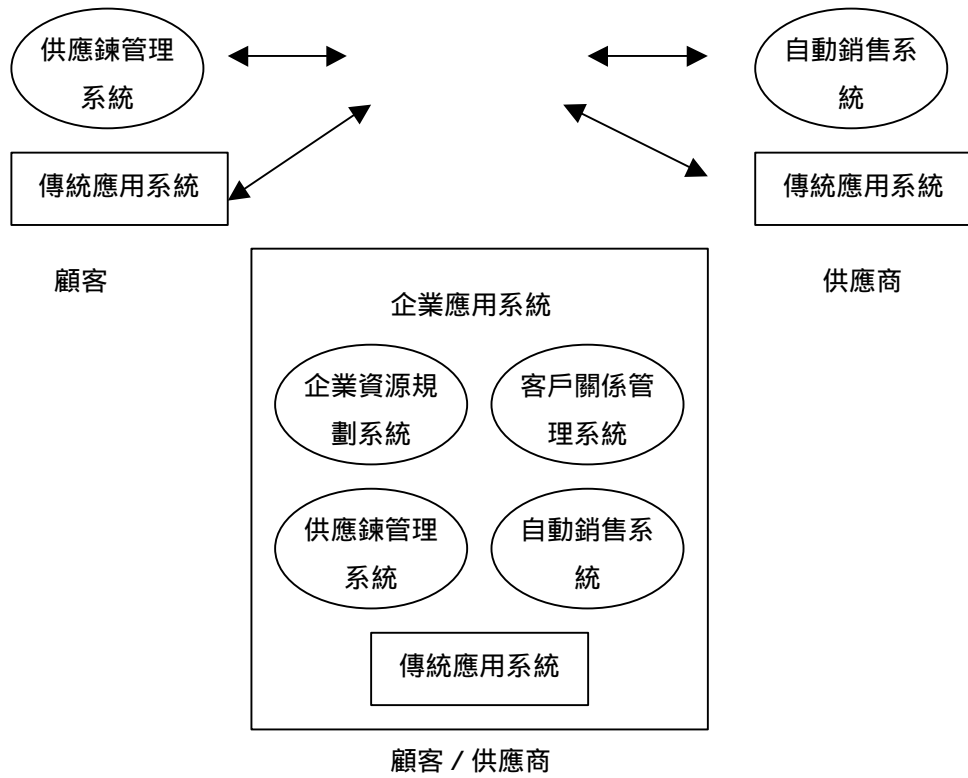


圖 2.7 電子交易市集整應用程式架構[5]

2.5 我國企業電子型錄應用及電子交易市集發展概況

電子型錄提供電子化的商品型錄資訊，可供用戶瞭解商品內容、透過電子郵件詢問商品以及獲取商品的即時交易資訊。電子型錄的電子化型式，包含光碟、單機電腦以及網路型錄服務。本文所言的型錄服務，皆是指網路型錄服務，它是電子型錄的主要型式。CommerceNet 在 1999 年十一月[5]，針對亞洲主要工業國台灣、中國大陸、日本、南韓以及新加坡，這五個國家各類產業的廠商進行供應鏈調查，調查的目的在查驗各國應用供應鏈管理範疇中的電子型錄的狀況。抽樣有效樣本為台灣 82 家、中國大陸 153 家、日本 166 家、南韓 81 家、新加坡 100 家廠商。其中日本及南韓 80% 以上為年收入在 1 億美金的大型公司，中國大陸為年收入 1 千萬美金以下的小型公司，而台灣及新加坡則大部分為介於二者之間的中型公司。調查結果發現台灣當時有 6 成以上的公司提供電子型錄服務，此比例在五國中最高，遠高於只有 3 成不到的中國大陸。所有受訪公司的直接性物料供應商則平均有 55% 提供電子型錄服務，五國在這地方則無多大差異。在問及使用電子型錄目的為何時，台灣有 55% 公司是為了參考及線上採購，此比例在五國中最高。反而亞洲經濟最先進的日本，利用電子型錄的目的僅作參考目的，採購則是依靠傳統的電話及人員接觸。所有受訪的公司提供的電子型錄有三成以上具備產品價格、產品規格書以及線上採購功能，只有 4% 具有議價功能。普遍來說，只有四成左右的受訪公司曾經透過網際網路進行線上採購，造成不想使用線上採購的主要因素是資訊安全以及無法進行線上議價。相對來說，其實台灣的公司有很高的意願從事線上採購，唯台灣有為數眾多的中小型零件供應商，所以對商品型錄搜尋有其必然的需要，才能在眾多的供應商中找到最佳的合作夥伴。

企業尋找供應商作為交易對象的最佳解決方案，就是透過電子交易市集的即時電子型錄服務。根據資策會 MIC 的報告指出，目前國內的中小企業業者對於

電子交易市集的功能需求，停留在基本的訊息資料與廠商搜尋上，亦即對於電子交易市集的需要僅在電子型錄提供上。如果以第三節的分類標準，台灣的電子交易市集主要的經營型態是型錄服務模式，本身不提供交易功能。主要的原由即在同性質的產品，在國內即有多個供應商可作為交易對象選擇，所以企業採購商品面臨最困擾的問題即在尋找供應商。此外，供應商對電子交易市集滲入其商品交易行為，以獲得交易仲介費，普遍不表贊同。所以一個能夠提供完善型錄增值服務的電子交易市集，將是顧客以及供應商皆樂見的。

在美國許多產業皆開始發展各自的電子交易市集，在台灣由於在全球電子產業供應鍊扮演重要角色，特別是在資訊電腦產品上。所以目前全球最大的二個電子產業電子交易市集，將台灣視為發展亞洲電子產業電子交易市集的主要根據地，關於電子產業電子交易市集相關議題，將在第七章中作更進一步詳述。除了電子產業之外，台灣是全球第十三位的貿易國，其他產業在世界亦扮演同等重要角色。為求台灣的重點產業可以導入電子交易市集，為台灣廠商尋找國外客戶，所以經濟部工業局積極進行「台灣產業全球電子市集整合發展及聯盟合作推動」計畫，簡稱台灣產業市集 TIM(Taiwan Industrial Marketplace)計畫[36]，並於 2001 年 3 月 12 日舉行 TIM 論壇成立大會暨第一次聯合會。這個計畫預期整合國內十大產業，包含紡織、石化、食品、機械、資訊、車輛、鋼鐵、造紙、電子電腦及一般產業。預計在 2001 年 5 月推出整合性的入口網站，三年內將達成 1 萬家企業，30 萬件商品上線登錄，以及至少 10 萬筆交易的目標。由政府部門主力推動的 TIM 是國家規模的電子交易市集，在參與廠商及商品型態數量上，將可以與目前最具跨產業規模的 VerticalNet 電子交易市集相較競爭。

要完成 TIM 必須先在產業體系內透過資訊交換標準進行整合，在跨產業體系外透過流程整合標準進行整合，最後建立產品分類標準，透過 TIM 入口網站介面以供國外客戶、國外電子交易市集或是海外台商，查詢廠商資料、詢價、議

價、採購、轉帳等，並提高台灣產品在國際市場的能見度，增加各國對我國的採購量。關於 TIM 的架構，以圖 2.8 加以表示。

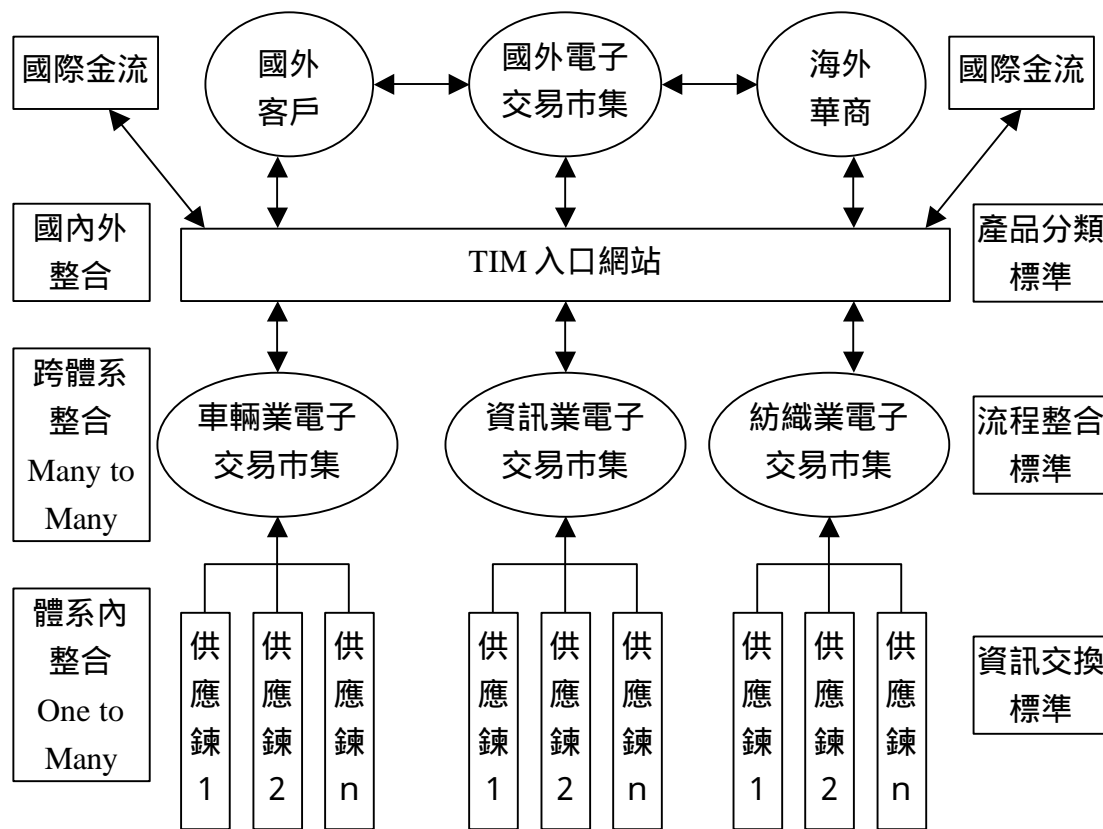


圖 2.8 TIM 整體架構[36]

TIM 對將來我國製造業的發展有非常重要的影響及貢獻，TIM 的目的在作為台灣製造業的型錄服務仲介。在每個產業中，皆有各自的電子交易市集運作，而產業內的資訊交換標準，亦是一種趨勢。一個扮演公正第三者的公眾式電子交易市集，是能夠吸引產業內多數企業參與的市集，如果說一家企業未能提供符合該產業資訊交換標準的型錄內容，則該電子交易市集有能力且有責任，將參與企業提供的型錄或是顧客的型錄請求資訊，轉換成符合產業資訊交換標準的文件。足見公眾式電子交易市集，是達成產業體系內整合的重要利器，在國內其發展將受到政策相當的鼓勵。目前國內的電子交易市集亦多屬公眾式電子交易市集，包括貿易風、Global Sources、台灣商業網以及亞普達等。它們提供的服務亦未實際介入買賣雙方的交易行為，僅提供佔多數中小型供應商的型錄服務。

2.6 小結

2.6.1 型錄服務模式的電子交易市集理想經營策略

運作型錄服務模式的電子交易市集，通常是公眾式的電子交易市集由公正第三者所主導，而且跨數個產業領域，最著名的例子是 VerticalNet。本論文所設計的类型錄搜尋代理人架構，即欲應用在型錄服務模式的電子交易市集。型錄服務模式的電子交易市集，隨著各國商業文化背景不同，相信型錄服務模式的電子交易市集，其延伸出的細項類別將是形形色色、各具特質。所以，經營一個電子交易市集可以說相當複雜。筆者認為經營此模式的電子交易市集業者，應該與外部企業尋求合作的方式來經營電子交易市集，建議應該專注在商品型錄的加值服務，不宜多角化經營。電子交易市集可以協同合作的企業包含：

1. 電子交易市集技術供應商：可以支援電子交易市集相關技術支援的廠商皆包含在內。包含早期進入電子交易市集領域，並提供採購自動化軟體的 Commerce One 及 Ariba，提供作業平台支援的 IBM，資料庫服務的 Oracle，供應鍊管理軟體的 I2，利用 XML 作為系統整合技術的 WebMethod 等。

2. 專業顧問公司：一個從事特定產業的垂直式電子交易市集，如果市集平台的研發團隊不具備該產業的領域知識背景。則電子交易市集平台將無法滿足產業中各個廠商的需要，無法吸引用戶進入該市集。所以向具備特定產業專業知識的顧問公司諮詢，將有助於電子交易市集的網站平台，往正確的方式經營。

3. 制定電子商務標準的組織：隨著各個領域的電子商務標準的建立，電子交易市集應遵行該領域標準，才可以與各企業的系統相容。

4.數位憑證授權單位：雖然電子交易市集本身可以當作市集參與廠商的數位憑證授權單位，但是光是市集授權的憑證並沒有足夠的公信力。所以可以尋求與專業提供數位憑證服務的 VeriSign 合作，讓市集本身具備安全可靠。

2.6.2 電子交易市集未來趨勢

電子商務的發展，從最先的企業對顧客(Business to Consumer)模式，簡稱 B2C；到後來的企業對企業(Business to Business)模式，簡稱 B2B。即使專業分析單位對 B2B 採取的樂觀態度遠大於 B2C。但是 B2C 的模式，相信不會就此一蹶不振。事實上，一個企業對企業對顧客(Business to Business to Consumer)，簡稱 B2B2C 的觀念，逐漸被業界所採納。使用 B2B2C 的模式，企業可以接取消費者訂單後向供應商自動調貨，達到零庫存即時生產的目的，這對存貨成本高的產品尤其重要。筆者認為，雖然電子交易市集相關資訊多數來自於探討 B2B 的文獻。但是經過仔細思考，其實 B2B2C 的模式，更適用在電子交易市集之中。電子交易市集匯集了許多的廠商，這些廠商的顧客群有些來自企業，有些則來自消費者。電子交易市集更能提供詳細供應鍊上的廠商彼此的商品資訊，比起原先存在於廠商的供應鍊軟體所存有的商品資訊更加豐富。比如一個匯集手機零件供應商、製造商、銷售代理商、消費者的電子交易市集，即能替手機業者提供 B2B2C 模式的運作。因為 B2B2C 的模式出現，將會使得傳統 B2C 及 B2B 的界定更加模糊。電子交易市集將會主宰這個模式的進展，並成為未來電子商務運作的主要模式。

另一方面，目前的電子交易市集在供應鍊整合上扮演重要的角色，但僅有助於企業在原料資源的規劃。未來電子交易市集持其扮演企業虛擬社群凝聚者的角

色，必能促進完成企業之間從供應鍊整合走向企業資源規劃的整合。

第三章 行動代理人架構

在現今網際網路的時代中，線上的資料資源淋榔滿目，想要透過網路來尋找對使用者有用的資訊，即使利用現今的搜尋引擎，能夠找到的不是重覆出現過多就是令人失望的不相關資料，造成蒐集資訊相當大的困擾。在電子交易市集中，如果說只是由眾多的電子商店聚集在同一個入口網站中，那麼這個市集對使用者的受益並不大，因為該市集無法有效地提供市集用戶有用的資訊。幸運的是行動代理人技術，替我們解決了這部分的問題。

行動代理人可視為具有軟體代理人特質的一種分散式運算架構，具備有代理使用者執行工作功能及行動運算能力的二個主要特徵。在探討軟體代理人系統及分散式運算架構的文獻中皆會列示行動代理人系統，基於行動代理人在這二個領域的特性，將在第一節及第二節詳加描述。第三節對行動代理人的優點及架構運作模式評估，最後第四節作本章的小結。

3.1 軟體代理人分類與比較

何謂軟體代理人，在學術上並沒有給它很嚴格的定義，在各類書籍及書出版物都對軟體代理人有大同小異的定義。利用軟體代理人技術可以應用在工作流管理、電信網路管理、交管控制、企業程序再造、資料探勘、資訊管理、電子商務、教育、個人數位助理、電子郵件過濾、數位圖書館、命令控制、智慧資料庫、排程管理等情境。預言在十年之內許多消費品都會內建代理人系統。

Nwana, Ndumu 在”Agent Technology”一書中將軟體代理人可分為七種型式

[16]：協同式代理人(Collaborative agents)、介面代理人(Interface agents)、行動代理人(Mobile agents)、資訊網路代理人(Information/Internet agents)、觸動式代理人(Reactive agents)、複合式代理人(Hybrid agents)、智慧型代理人(Smart agents)。茲將這七種型式的代理人的性質列於表 3.1 作一比較。雖然各類型的代理人設計的目的不盡相同，但是它們有一致的特徵，它們存在一個環境之中，具備能力與它們的執行環境互動，並且非同步化且自主性地在該環境中運作。不需要人為的介入，代理人便能夠簡單地持續實現它的運作目的。從最終使用者的觀點來看，代理人是一個幫助使用者並且扮演他們身分完成派定工作的程式。從系統觀點來看，代理人是一個軟體物件，它存在一個執行環境之中，擁有自動性、目標導向、感應事件、持續執行的必備特徵，以及通訊、行動、學習、取得信任的擴充屬性。

行動代理人是軟體代理人中的一種，它具備有行動性是軟體代理人的擴充屬性之一，亦即並不是所有的軟體代理人皆具有行動性，具有行動性的代理人，稱之為行動代理人，而不具行動性的代理人，可廣泛稱之為靜態代理人。所以，除了上述的標準來分類軟體代理中之外，亦可將軟體代理人依照其是否具備行動運算能力，而扼要地將其分成行動代理人與靜態代理人二種。靜態代理人只能在起始的平台上執行，如果它需要資訊來進行執行時，傳統上可以透過遠端程序呼叫與外界的不同系統進行通訊。而相對的，行動代理人就沒有被限制在起始的平台上才能執行。行動代理人可以自主旅行在網路上的各主機，它可以在一個執行環境中被起動，然後透過網路載送它的程式碼(code)及狀態值(state)送至另一個執行環境，然後在那裡持續執行尚未完成之工作。上述所指的程式碼，指的是物件導向環境中代理人執行必備的類別碼；而狀態值，指的是代理人的屬性值，可以幫它決定當它到目的地繼續執行後要做什麼事。

類 型	性 質 分 析
-----	---------

協同式代理人	強調自主及與其他代理人合作，因應分散式人工智慧目的而設計。解決中央集權式大型系統的缺點，分散成許多代理人再行協同合作可以降低運作風險，並且可以與原有作業互通。繼承人工智慧的特性，可以在知識層次上運作，且容易維護，並兼具再使用性及平台獨立等優點。
介面代理人	強調自主及學習能力，核心能力是成為使用者的個人助理。合作的對象是使用者而非其他代理人，是可以協助使用者與電腦溝通的介面軟體。透過觀察倣效使用者，接受使用者回饋，接受指令及詢問其他代理人建議的方式，來學習如何幫助使用者。因為它強調容易使用，及在專業領域中運作及不需要其他軟體合作，所以預期它會很快被快速接納。
行動代理人	行動代理人是本章研究的核心，將在本章第三節作更充足的介紹。簡單來說，它是一個可漫遊在網路環境中的軟體程式，此程式由來源主機送至目的主機，代表它的使用者在目的主機完成派任工作，並將執行結果帶回來源主機。目前行動代理人應用雖不充足，但是預期未來將被電子商務領域廣泛應用。
資訊／網路代理人	資訊代理人扮演對許多分散在各地的資源進行資訊的管理、操作、校對等工作的角色。發展此代理人的動機在管理全球資訊網帶來爆炸的過多資訊，因此它強調可以幫助瀏覽器用戶更容易清楚的掌握相關資訊。未來資訊代理人將會走向在本地端建立個人資訊庫型式，讓使用者更能掌握有用的資訊。
觸動式代理人	觸動式代理人是一個在它們所處環境中沒有內部實作程序的代理人，此代理人受到所處環境的現行狀態改變的刺激而做出特定的反應。建立此代理人的核心想法是急切的理由，而此代理人在設定之初即沒有預設的行程，所有的行為都是外界的刺激而產

	<p>生。另外，為求快速反應，此代理人會將工作進行分解成數個模組，每個模組單獨執行，並要求其通訊量達至最小，以使它可以快速完成反應行為。使用此代理人最大的好處在於其執行穩定且容錯能力佳的本質，以及高彈性及適應環境能力。</p>
複合式代理人	<p>複合式代理人是結合上述五種中的二種或二種以上的代理人。</p>
智慧型代理人	<p>兼具有協同式代理人的協同合作天性，加上具備介面代理人學習環境的本質，智慧型代理人具備自主學習及協調其他代理人完成派任工作的能力。</p>

表 3.1 軟體代理人分類與比較[16]

3.2 分散式運算架構分類與比較

3.2.1 分散式運算架構

在現今分散式運算的環境中存在許多不同作業平台的電腦，串連在網際網路的領域之中，為因應異質的分散式系統可以建立彼此互通的機制，因而發展出各式各樣的分散式運算架構。Ghezzi, Vigna 在“Mobile Code Paradigms and Technologies: A Case Study”[4]一文中將分散式運算架構分類成四種：主從式架構(Client-Server Paradigm)、遠端計值架構(Remote Evaluation Paradigm)、程式碼隨選架構(Code On Demand Paradigm)及行動代理人架構(Mobile Agent Paradigm)。假設在分散式架構中有二個作業平台，一個稱為 A 及 B 平台。分散式運算架構基本的差異在執行應用所需要知識(know-how)、資源(resource)及處理器(processor)初始存在平台不同。其中知識是執行運算所必備的程式碼，資源是運算所需的輸出入資料，而處理器是執行程式碼的虛擬機器，即程式碼的直譯器。換言之，我們也可以把知識當作是解決問題的方法、步驟，資源是問題本身，而處理器是解決問題的工具。必須將知識、資源及處理器放在同一作業平台上，才能執行運算，解決問題。所以知識、資源及處理器是分散式運算的三個重要元素。這些運算架構有主從式架構、遠端計值架構、程式碼隨選架構及行動代理人架構四種，這些架構依照其初始存在這三種元素的平台不同，及執行運算的平台不同而相異，可以表 3.2 來表示。為了要清楚分析各類架構的不同，列舉一個分散式資訊採集系統作為實例加以說明。這個系統是由一些資料庫管理系統及一個搜尋引擎所組成，每個存在網路上各主機的資料庫管理系統負責維護各自的資料表，這個資料表由關鍵字及資料二個欄位所組成。關鍵字是資料的索引值，而資料是存放與此關鍵字相關的資訊，如果此主機未存有此類資訊，就會在資料欄位內記錄存此類資訊的其他資訊庫位置。而搜尋引擎的工作是蒐集大量與此關鍵字相關

的資訊，本身具備使用者介面以及與資料庫管理系統連繫的機制。現在將上述的分散式運算架構加以簡介並應用此案例加以引述，如下所示：

分散式運算架構	A 平台	B 平台
主從式架構	-	知識、資源、處理器 *
遠端計值架構	知識	資源、處理器 *
程式碼隨選架構	資源、處理器 *	知識
行動代理人架構	知識、處理器	資源 *

*表示知識、資源、處理器一起執行的地點[4]

表 3.2 分散式運算架構

1. 主從式架構(Client-Server Paradigm)

將系統分成用戶端以及伺服器二個固定角色，A 平台即用戶端本身不具備處理運算以解決問題的知識、資源、處理器，所以向已知存放此運算功能的 B 平台即伺服器提出服務請求。這裡上述案例的資料庫管理系統扮演伺服器的角色，搜尋引擎扮演用戶端的角色。這時存在用戶端的搜尋引擎對特定的資料庫管理系統提出關鍵字，作為向資料庫管理系統搜尋的參數，然後資料庫管理系統回應其相關資訊，並將存有此類資訊的其他資料庫管理系統位置告知 A 平台上的搜尋引擎，欲完成所有資訊的採集，搜尋引擎必須再依照此位置繼續查詢，直到所有的相關的資訊皆搜尋完成。

2. 遠端計值架構(Remote Evaluation Paradigm)

A 平台擁有處理運算必備的知識，但是缺少資源及處理器，所以將此知識送至存有資源及處理器的 B 平台中進行運算。在這裡搜尋引擎所在的 A 平台傳送程式碼片段至存放此資料庫管理系統的 B 平台上。在此平台上提供此程式碼可以執行的直譯器，讓此程式碼可以載入執行環境並透過資料庫管理系統搜尋相關的資訊。等到完成它的工作之後，即將此運算結果送回搜尋引擎。這個結果含有其他有提供此相關資訊的資料庫管理系統的位置，預完成所有資訊的採集，上述的步驟必須持續重覆運作。

3. 程式碼隨選架構(Code On Demand Paradigm)

A 平台擁有處理運算必備的資源及處理器，但是缺乏知識，所以向存有知識的 B 平台下載知識，以瞭解解決問題的方法，以便在 A 平台上處理運算。這時在 A 平台上存有欲搜尋的大量資料，但是它不具備搜尋的知識來處理大量的資料。因此在 A 平台上的搜尋引擎向 B 平台下載知識，以配合原來 A 平台上的處理器對存在本地端上的大量資料進行運算處理。本架構雖然只要傳送容量小的知識即能完成運算，但是需要在本地存有大量的資料，應用上並不實際。

4. 行動代理人架構(Mobile Agent Paradigm)

A 平台雖然擁有知識及處理器，但是沒有資源可以在運算中作為輸入的資料。所以 A 平台將知識及處理器傳送至存有資源的 B 平台之中，並且在那裡執行運算。這裡搜尋引擎被序列化封裝，然後在網路上漫遊以蒐集既定關鍵字的資訊。一開始和遠端計值架構一樣，是將 A 平台的搜尋引擎程式碼移至存在資料庫管理系統的 B 平台，然後在 B 平台之中處理之後發現還有其他資訊庫管理提

供此資訊，它就會移至其他平台上再繼續搜尋相關資訊。為了防止同樣的平台被重覆的查訪，所以這個程式會存有記錄來瞭解那些平台已經到訪那些還需要去查訪。等到所有提供相關資訊的平台皆已經到訪過，則將其處理後的結果送回 A 平台以得出最後結果。

綜觀各類分散式運算架構的比較，並應用相同的情境 - 資訊採集系統加以評估分析。可以發現，利用行動代理人架構，可以減少網路上的資料流及資料量。因為行動代理人程式必備自主能力，所以不需要人為介入或是送回來源地再繼續下一地點的傳送執行，因此自然減少了訊息流通的次數。另外，行動代理人架構在網路上傳送的主要是序列化過的可執行程式，其採集的資訊都是經過運算處理過濾過所以不會重覆，所以大量減少網路上流通的資料量。

除了上述的分類標準之外，我們亦可將分散式運算架構，大致分為傳統架構，及新式架構二大類[31]。傳統架構是始於 1970 年代直至今日，這個架構稱為遠端程序呼叫(Remote Procedure Calling, RPC)，所有電腦彼此之間溝通的訊息是程序執行的請求(request)或是回應(response)，請求是程序的執行參數，而回應是它的執行結果。而程序是存放在負責執行該程序的電腦內部之中。用戶端電腦提出程序的請求，而伺服器在接收此請求之後給予相關的回應。比如，當用戶端想要刪除超過二個月以前的檔案時，首先用戶端電腦會送出請求列示使用者的檔案的名字及日期列出來，然後分析那些檔案需要刪除，並將此請求送出，如果要刪除 n 的檔案，則這會傳送及接受 $2(n+1)$ 次的訊息才能完成整個程序。

新式架構則是近幾年被提出，是取代 RPC 的方案，稱為遠端程式運算(Remote Programming)。這個架構讓電腦不只可以呼叫遠端電腦中的程序，還可以提供自訂的程序以供遠端電腦執行。在遠端程式運算架構中，每個訊息包含了接收端電腦的程序以及該程序所需的參數。最重要的改良在於，傳送方電腦與接

受方電腦之間協定一個共通的語言，這個語言包含了讓程序可以決策、計算、修改狀態值及呼叫接收方電腦的程序的指令。將程序呼叫從遠端拉近到本地端將可以更有效率執行程序。這個程序以及它的狀態值由行動代理人來載送，可以在傳送方及接受方電腦環境中執行。同樣前述的刪除過期檔案的運作，此架構可以利用行動代理人來載送雙方可以理解的指令，行動代理人程式送至接受方電腦之後，即代理傳送方完成既定的刪除檔案動作，不需要再有其他的訊息傳送。而此架構的主要好處便是在不用提高網路頻寬的前提下，讓原先的工作執行的更有效率且穩定。

3.2.2 三層式 Web 平台

以上所述的分散式運算架構，是由二個或二個以上作業平台透過相同的通訊協定，即能替使用者完成特定的資訊服務。即使行動代理人架構是在多平台的環境中執行運算，但是最終來說它只要有二個平台便能實現其架構的運作。一個是使用者存取介面的平台，另一個則是提供資源或是資料庫的平台。因此這類的分散式運算架構，我們統稱為二層式架構。雖然二層式架構已經被產界所廣泛採用並開發應用程式，但是包含上述所列的架構皆必須面對下述問題[29]：

- 企業邏輯改變時須要重新建構用戶端應用程式。
- 存放在伺服器中資料庫，一旦其資料庫結構、資料來源有所更變，則用戶端應用程式也必須改變。
- 在用戶端平台的應用程式一旦改版，則所有的用戶端皆需要同步更新。
- 用戶端必須安裝連結資料庫所需要的驅動程式，並做相關介面設定。
- 過多種類的資料庫將使用戶端應用程式必須設定多種連結介面。

在談論資訊系統資源的執行權管制時可以發現，如果系統管理者為每個使用者建立帳號的方式，來管制其使用資源的權限。我們會發現當有新進員工加入或是員工職務變遷時，如果加上公司系統資源的種類又相當繁多，那麼系統管理者在管制使用者執行權限的工作會變成相當繁雜。而以職務為基礎的執行權管制，加入職務在使用者與資源權限之間，使得系統資源的管制變成相當有彈性。其中職務與權限的關連是企業既定的安全政策，必須透過對企業程序的瞭解來建立，所以期初成本較高。但由於員工與職務的關連簡單，所以日後系統管理者對員工的執行權管制，將只局限在員工與職務關連的建立，所以具備相當的彈性，而且維護執行權管制的成本將相形降低。

回到分散式運算架構的主題，當二層式架構常會面對上述問題很難解決時，其實可以加入一個層次，我們稱為中間層或是企業邏輯層，將用戶端應用程式與資料庫管理的程式分開。加入這個中間層後的架構，稱為三層式架構，由於中間層可能不只一層所以亦層為 N 層式架構。與前者所提以職務為基礎的執行權管制概念相同，加入本架構的中間層，必須對企業程序的瞭解來建立相關的物件模型。擁有企業邏輯的中間層必須與企業所有的資料庫建立關連，使其可以獲得所需的資源，所以期初投入成本較高。但是由於此架構的用戶端應用程式只是扮演中間層的使用者介面，所以關係簡單。日後一旦企業程序改變時，將只需要針對中間層的程式進行改版，而用戶端應用程式改變的機會不大。所以三層式架構的優點即在提升系統的彈性，並降低維護的成本。

由於全球資訊網的出現造就現今網際網路的快速發展，使得 Web 成為現今資訊系統開發的主流平台。企業的應用系統也逐漸放棄傳統架構改採以 Web 為基礎的平台。構成 Web 平台的主要二個元素是 Web 瀏覽器以及 Web 伺服器。在 Web 平台發展之初由於沒有資料庫連結機制，所以 Web 平台是建立在主從式架

構上，而後隨著 CGI(Common Gateway Interface)的出現，使得 Web 伺服器有連結資料庫的能力，而呈現目前以 Web 為基礎的三層式架構。其後，為了達到特定的應用，使得 Web 平台亦用在其他分散式運算架構上。比如從 Web 伺服器下載至用戶端瀏覽器執行的 Applet 是應用在以 Web 為基礎的程式碼隨選架構上；而 IBM Aglets 行動代理人系統中，內裝 FijiApplet 在用戶端的瀏覽器之中，使得用戶端派遣的行動代理人，可以透過存在 Web 伺服器中的 Aglet Router 轉送至目的地。

以 Web 平台為基礎建構的分散式運算架構，最大的優點是它建立一套所有用戶端瀏覽器皆懂的語言 - HTML。如此可以使得用戶端應用程式，可以容易與不同的以 Web 為基礎建構的資訊系統建立互通，讓存在用戶端的應用程式可以穩固不變。另外，個人電腦作業系統的全球市場佔有率最高的微軟公司，已經將其發展的 Web 瀏覽器放在作業系統之中，足見 Web 平台是未來資訊系統建構的趨勢。因此設計以 Web 為基礎的應用系統，將可以使該資訊系統迅速被用戶取用接受。加上 HTML 文件的易讀性高，使得系統維護極富彈性。

3.3 行動代理人架構優點與系統運作模式

3.3.1 行動代理人架構的優點

從實際應用的角度來看，我們會自問為何要使用行動代理人架構，它在應用上有何利基，足以吸引社會來接受投入。Lange,Oshima 在 “Seven good reasons using mobile agent” 一文中將為何要使用行動代理人系統的理由分述如下[7]：

1. 降低網路上的資料負載：將資料在遠端主機進行運算，然後將結果送回本地主機，這是行動代理人運算求值的方法。比起主從式架構，是將大量資料從伺服器送到用戶端執行運算，或是其他架構需要數次的訊息往來才能達到一項任務，利用行動代理人系統確實降低了現今網路人口增加而造成的網路擁塞問題。又因為此系統減少網路傳輸的流量及次數，所以在不安全的網際網路環境中，更增添一層安全的保障。
2. 克服網路延遲問題：在敏銳性高的即時系統中，例如從事製造工作的機器人，需要即時反應它們目前的環境。由於這類即時系統需要頻繁的訊息傳遞才能即時反應目前狀態，而如果因為資料量過大而造成訊息的延遲，是這類系統不能接受的。而行動代理人提供了一個這個問題的解決方法，因為中央控制台可以將解決問題的知識，從中央控制台被派遣到本地平台上執行，因此解決過多訊息通訊而可能造成的延遲問題。
3. 在目前通訊協定上層實作：在分散式系統中當要進行資料交換時，每部主機必須擁有各自的程式碼來實作進出的資料，才能進行資料通訊。行動代理人可以與遠端主機在現有的通訊協定上層，建立通道來進行資料通訊，因為在現有協定上實作所以在程式開發上更有效率，並且可以運用該協定上已經發展成熟的安全防護機制，所以更有安全保障。
4. 非同步且自主的執行：觀察目前用戶人口迅速增加的行動通訊產品，可以印證商業社會對行動通訊的蓬勃需求，但是目前行動通訊的資料傳輸頻寬低加上連線穩定性差，所以許多通訊廠商及組織不斷在技術上改良，以尋求突破。儘管如此，近幾年內行動網路的通訊品質必定會與固接網路有一段不小的落差。除了品質考量之外，成本亦是重要因素，目前行動網路的費用比固

接網路高出許多。所以在行動網路上進行資料交換，我們期待可以用最小的資料傳輸來達到不變的目的。而行動代理人因為可以讓使用者透過行動網路派遣代理人之後，即可切斷連線，然後等到該代理人在遠端主機上自主地執行並且產生結果之後，才再次通連網路將結果送回使用者的行動設備，達到非同步傳輸的目的。

5. 動態的環境適應：行動代理人可以感應外在的執行環境並自主地調適其行為。在分散式系統中，多元行動代理人的設計，可以將多個代理人分散至各主機中，以最佳化的組態配置來解決特定的問題。
6. 具有異質系統調適的本質：網際網路透過共通的協定來進行全球電腦的通訊，但是電腦系統從作業系統或是硬體的觀點來看，它們基本上是不同性質的。相同的軟體不易在不同的電腦硬體或是不同的作業系統上皆能運作。而行動代理人是建立在傳輸層之上，它只依賴所處的執行環境，因此行動代理人可以說提供最佳的條件讓系統進行整合。
7. 功能強固且容錯性高：行動代理人具備感應周遭不利環境並自主地作出即時反應的能力，使用利用它可以很容易地建構功能強固並且容錯性高的分散式系統。假如存在一部主機中的代理人，發現它所處的主機將要關閉，那麼代理人系統將會通知所有存在這部主機的代理人，然後派遣代理人至其他的主機執行其預定完成的工作。

比較前述各類分散式運算架構比較及前述的使用行動代理人的七個理由之後，可以歸納出行動代理人架構在量與質上都有不小的改進。在量上的改進是遠端程式運算架構比遠端程序呼叫更有績效。用戶端電腦派遣行動代理人至遠端伺服器之後，即直接在此伺服器上進行執行派任的工作，不需要有額外的訊息傳

遞，所以執行工作更有績率。在質上的改進是遠端程式運算架構可以客製化。行動代理人架構激勵用戶端軟體的製造商，將原用戶端軟體增加伺服器軟體所具備的功能。如上所述，如果檔案伺服器的軟體提供列示使用者檔案目錄及刪除檔案的功能，那麼用戶端也可以利用其用戶端軟體，有效地增加一個刪除所有過期檔案的新程序。這個新程序由代理人的型式組成，為特定用戶端提供客製化它的資訊服務。

3.3.2 行動代理人系統的運作及通訊架構

行動代理人系統掌控代理人在其生命週期內的所有運作，這些運作包含行動代理人的產生、複製、派遣、喚醒、暫停、召回及移除等。另外，前一節提到行動代理人系統中的主要元素之一的代理人擁有與外界溝通的介面，在此將對行動代理人系統的通訊架構作進一步探討，以瞭解行動代理人系統如何運作，使得代理人之間或是代理人與系統之間可以作訊息交換。首先，探討行動代理人系統的運作架構，我們可以將代理人系統的運作想作是代理人從產生至移除的執行生命週期，這個週期包含下面幾個階段：

1. 產生(creation)：代理人的產生是發生在執行環境之中，新發生的代理人被指定一個識別標記，進行執行環境中並且在此初始化。
2. 複製(cloning)：產生與原代理人幾乎相同的備分，唯一不同的是它們的識別標記不同，而且複製的代理人必須重新進入執行狀態。
3. 派遣(dispatching)：將代理人從一個主機的执行環境之中移除，並將它送到目的主機的執行環境，在那裡重新執行。
4. 召回(retraction)：將代理人從現在所處的執行環境中拉回至行使召回命令的

執行環境中。

5. 暫停及喚醒(activation and deactivation)：暫停是代理人停止目前的工作，將它存在當地主機的儲存體之中呈睡眠狀態，而喚醒是指將在儲存體中睡眠的代理人叫醒，使它進行待命執行狀態。
6. 移除(disposal)：代理人停止目前的執行活動，並從現在的執行環境中清除。

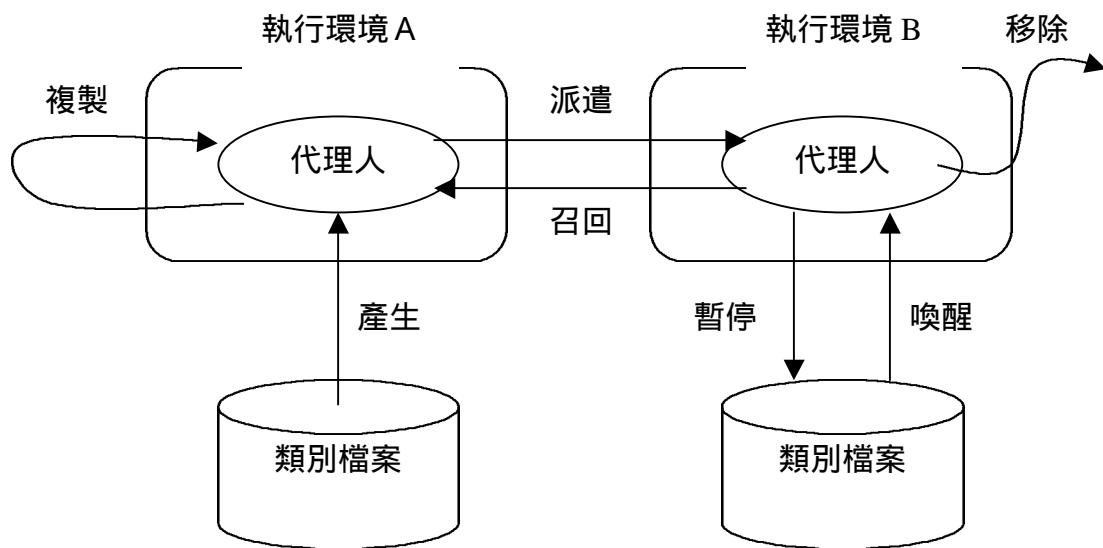


圖 3.1 行動代理人生命週期[6]

其次，我們探討行動代理人系統的通訊架構。行動代理人透過代理人系統的通訊機制可以與同一執行環境或是外部執行環境的其他代理人進行通訊。代理人可以運用其他代理人的物件方法或是傳送訊息給它，而傳送的方法可以是一對一或是一對多的。代理人之間的通訊使用下列三種傳送方法：

1. 即時型傳訊(now-type messaging)：利用同步傳輸的原理，訊息發送方送出訊息往收受方之後，即停止目前的工作直到接受方完成處理回應訊息。
2. 未來型傳訊(future-type messaging)：利用非同步傳輸的原理，訊息發送方送出往接受方的訊息之後，不會停止目前的工作，並且保持對此訊息的控管，

以隨時接收來自對方的回應訊息。這個方法提供在多元代理人通訊上很多的彈性。

3. 單向型傳訊(one-way messaging)：與前者相同，這個方法也是利用非同步傳輸的原理。不同的是傳送方送出訊息之後，不會保持對此訊息的控管，所以不等待接受方的任何回應。

3.3.3 行動代理人系統的設計架構

將行動代理人系統來建構一個完整的資訊系統，會因為應用情境不同，而有不同的設計架構來因應。這裡舉出常見的三種架構：

1. 主從式架構：一個主要的代理人可以開啟一個或數個附屬代理人，然後指派這些代理人特定的任務。基於效率的理由，主代理人開啟數個代理人之後，可以採平行處理的方式將這些代理人同時派遣至遠端平台，等到執行任務完成之後，將結果傳回給主代理人，然後自動在遠端平台中移除自己。這類架構的應用情境是一個任務的執行需要許多工作同時處理，或是靜態代理人想要在遠端執行任務。例如：在需要資訊推播的軟體更新服務應用系統，適合採取這類架構來促進軟體元件的即時安裝。
2. 行程式架構：行動代理人參考它的行程物件，以決定其下個目的主機，使得這架構上的行動代理人可以遊走於網路上的每部在預定行程中的主機。當行動代理人依照行程物件派遣至各地主機時，如果發現找不到該部主機，它會嘗試當自己派遣到另一個新的目的地。依照各地情況的不同，它會動態地將

行程作必要的修改。這類架構的應用情境是預先作行程規畫，讓行動代理人可以重複利用及分享。例如：電子公文系統由於需要既定的程序才能完成公文的傳遞，所以適用這類行動代理人系統的設計架構。

3. 以行程為基礎的主從式架構：同樣是利用上述第一種主從式架構，我們可以將此架構加入行程控制的機制，而轉變成第三種行動代理人系統的設計架構。主從式架構是利用平行處理的方式將代理人同時散佈至各地主機，而此架構是主代理人產生附屬代理人之後，派遣在外的代理人參考其動態的行程，以決定行徑路線。當代理人移至遠端主機後即執行其指派的工作，然後將執行結果送回主代理人。這類架構是參考行程式架構的主從式架構改良模型。這架構最大的優點有二個，第一個優點是它不會像主從式架構般浪費網路頻寬資源，因為主從式架構是平行處理，所以如果目的地是 N 部主機，那麼主代理人必須產生 N 個附屬代理人，再將這些含有程式碼的代理人送至遠端主機之中。因為這個架構只需要主代理人產生一個附屬代理人，就可以在 N 個主機中行走，比起前者相當節省網路頻寬。另外一個優點是它不會像行程式架構，代理人必須將各地執行結果存在內部，如此蒐集的資訊容易遭到別的主機的竊取。因為這個架構是隨時將派遣在外的附屬行動代理人執行的結果，送回至主代理人，所以附屬代理人不需要將各地主機所執行的結果存放在自己內部，而保障此資訊不會一併遭竊取。在應用上它適合用在本文將實作的電子交易市集中的商品型錄服務流程。

3.3.4 以 Java 程式語言開發的行動代理人系統

本節開始說明行動代理人系統是實作行動代理人應用系統的語言，目前存在有許多行動代理人系統來開發以行動代理人為基礎的應用系統，這些行動代理人系統大致可以分為以 Java 程式語言開發的行動代理人語言，以及非利用 Java 程式語言開發的行動代理人語言。先談後者，利用非 Java 的程式語言來開發行動代理人系統，缺點自然是缺乏 Java 執行環境(Java Runtime Environment, JRE)的相容。這些行動代理人系統包含美國 General Magic 公司開發的 Telescript 系統[10]，美國 Dartmouth 學院開發的 Aget Tcl 系統[8]，以及由德國 Kaiserslautern 大學開發的 Ara 系統[14]等，其中 Telescript 系統是第一個商務應用的行動代理人系統。這類系統是專為行動代理人建構的操作平台，雖然可以透過系統介面的開發來與其他程式語言(比如 Java)開發的資訊系統進行整合，但是如此的話將使系統相容性低且投入研發的成本高。相對地，利用 Java 為開發其行動代理人系統的語言，優點即是 Java 執行環境的相容。這類行動代理人系統包含由 IBM 東京實驗室開發的 Aglets 系統[6][17]，美國 General Magic 公司開發的 Odyssey 系統[21]，以及由日本三菱公司開發的 Concordia 系統[20]等。值得一題的是開發第一代 Telescript 行動代理人系統的 General Magic 公司，為何會重新創造以 Java 開發的 Odyssey 行動代理人系統，這類行動代理人系統的目的在於提供 Java 一個應用的介面(Application Interface, API)，使得利用 Java 開發的應用系統可以容易地取用其功能，使得應用系統具備運作行動代理人的能力。Java 被公認為網際網路的語言，足見其在網際網路的重要性。那麼到底 Java 有什麼開發行動代理人系統的優勢，以下列出 Java 內建適用於行動代理人系統的機制，加以說明：

1. 跨作業平台：經過 Java 編譯過的二進位碼，可以在安裝有 Java 直譯裝置的平台上進行執行，所以 Java 有跨作業平台的能力。行動代理人系統必須提供行動代理人程式遊走於網路上，而 Java 跨作業平台的能力，使得行動代理人程式可以在安裝不同作業系統的主機上執行。

2. 安全的執行環境：Java 的執行環境提供了安全的防護機制，任何沒有授與權限的程式都不能非法存取系統資源。由於行動代理人是在不同主機上執行，所以必須注意外界主機對行動代理人的威脅，或是行動代理人對主機的威脅。而 Java 提供了足夠的安全機制，在預防相關的威脅。
3. 動態類別檔載入：類別檔是 Java 執行時所需的程式碼，Java 是在執行期間才動態地將必要的類別檔載入執行平台，因此提高的程式的可用度及降低程式下載的體積大小。行動代理人系統派遣至網路上的代理人，是將代理人本身程式移至遠端主機上執行，而執行期所需的類別檔，可以動態地依照需要，從代理人來源主機或是第三者主機下載必要的類別檔。
4. 多元執行緒的程式：Java 可以開發多元執行緒的程式，執行緒是在程式執行過程中，能夠自主執行程式碼的執行單位。由於代理人是具有自主性的，這意味著在相同地點執行的多個代理人可以獨立作業。而 Java 不但是可以開發多元執行緒的程式，而且它還提供同步功能，使得代理人之間可以互動。
5. 物件序列化：Java 的物件序列化是指將用來描述物件的狀態值輸出至檔案的過程，經過序列化的物件被表示為一序列的位元組。行動代理人系統強調代理人的永續性，派遣至網路上的行動代理人必須永續存在，才能穩定地在各主機上執行，而 Java 的物件序列化機制，使得行動代理人物件本身可以不易遺失。

3.4 小結

行動代理人架構具備同步 / 非同步以及單點 / 多點的訊息傳輸性質，加上處

理運算的過程是傳輸程式碼，而非大量的資料。因而本架構應用在電子交易市集環境中，對多數採行撥接或非固定網路位址連線的中小型企業用戶來說，得到權益的保障。行動代理人架構持續在網路上派送執行的能力，適合應用在以採集資訊為目的資訊系統應用上。商品型錄服務是電子交易市集必備的資訊服務，而採集大量由供應商提供的商品型錄資訊，行動代理人架構提供最有效的方法。唯行動代理人雖然在資訊搜整上有強大的支援能力，但是要採集各廠商間充足的資訊，必須要求各廠商提供電子交易市集規範的商品資料庫格式。純粹使用行動代理人架構，必須要求各系統有相同的界面。雖然限定採行一致的界面，是資訊系統整合最快且有效的方法。但是在應用上卻限制各使用單位定義資訊的彈性，亦難與存在於各單位後端的資訊系統進行整合。所以在下一章中，將整理 XML 相關資料，在彌補行動代理人實際應用上的缺點。利用 XML 架構在行動代理人分散式運算架構之中，可以發揮資訊系統整合及運算效能提升的功效。

第四章 XML 架構

4.1 XML 簡介

「可延伸性標示語言」(Extensible Markup Language, XML)以下簡稱 XML，是一個可以自我描述的標示語言，應用程式可以透過它來描繪其表述內容，以期與其他應用程式溝通整合[30]。XML是由當初推動業界 HTML 標準的 W3C(World Wide Web Consortium)所制定的標準。當初 W3C 制定 HTML 標準是希望各廠商所研發的瀏覽器可以有一致遵循的準則，才不致同樣的文件內容在不同的瀏覽中有不同的展示結果。由於 Web 的盛行使得 HTML 所描繪的文件，亦即 HTML 文件，成為現今最普遍且易於用戶取得的資訊。但是 HTML 受制於結構不嚴謹以及組成元素無法自行定義，致使它只能作為瀏覽器解析的應用，不易作為其他應用程式解析的目的。根據全球電子商務專家的預測，XML 所具備的可延伸及自我描述性，從電子文件內容之收集、資料的處理解析、儲存、自動傳輸，乃至後端的資料搜尋比較、運算、再處理和呈現，幾乎完全可以由電腦代勞[33]。本節針對 XML 作簡述，第二節則對 XML 如何作為訊息交換平台以達到系統整合的目的，第三節討論各界所制定的 XML 相關商務標準。

4.1.1 XML 的發展演進與文件架構

XML 及 HTML 皆源自於 SGML，SGML 在 1986 年被制定成文件標示的國際標準。第一個 HTML 規格是由 W3C 在 1992 年公佈成為 Web 網頁的標示語言。目前已經發展至第四版。而 XML 是在 1997 年開始被該協會討論，第一版是在 1998 年 2 月推出，被現今的 Web 瀏覽器所支援。因為 HTML 有固定的標示(tags)組成文件語法，再加上容易加入多媒體物件及文件的互相鏈結能力，使得 Web 平台能夠快速地在近幾年被大眾所採用，成為現今資訊系統開發的主流平台。但

是 HTML 最大的缺點即在使用者無法自訂文件中的標示，因而缺乏應用的彈性。如果以 HTML 文件作為資訊蒐集系統的搜尋來源，將很難被不同資訊平台所解讀而達到資訊蒐整的目的。起因於 HTML 語法結構不嚴謹而且此語言無法自行定義使用者標示，所以在此基礎上 Web 應用程式很難達到互通性。

相反地，XML 標準本身具備上述原創者的建構目標，與 HTML 最根本的不同在於 XML 可以擴充 HTML 無法自訂的文件標示。XML 創造出比 HTML 更豐富文件描述結構，因為應用上的彈性所以 XML 被認為是下一代的文件標示語言。同樣由標示語法組成的 XML 文件，不同於 HTML 文件的地方在 XML 文件的標示除了具有使用者定義的標示之外，還有其嚴格的標示文法要求。符合 XML 文法規範的文件稱作合於文法的(well-formed)XML 文件，而符合資料型態定義(Data Type Definition)的文件則稱作有效(valid)的 XML 文件。前者是要求文件符合 XML 標準的實體結構限制式，例如標示的區間比對。後者是要求文件符合 XML 標準的邏輯結構限制式，即符合使用者自訂的資料型態定義(Data Type Definition, DTD)。一個有效的 XML 文件必定是合於文法的 XML 文件，而一個合於文法的 XML 文件未必是一個有效的 XML 文件。



4.1.2 XML 應用範圍

由於 XML 具有強大以及高延展性的優點，因而使得人類開始思考它可以應用的情境。以下列出 XML 主要可以應用的三大範圍[13]：

1. 使用 XML 描述關於文件或是線上資源的上層內容(meta-content)：XML 一開始主要被視為上層資訊的語言，上層資訊是記載標題、作者、檔案大小、開啟時間、修正記錄、關係字等文件內容的資訊。上層資訊可以用在資料搜尋、資訊過濾及文件管理的應用上。在搜尋應用中，文件的上層資訊表達十分重要。在現今的搜尋引擎中，我們應用關鍵字所找到的資料，可能充滿太多不相關資料，起因在於 HTML 本身描述資訊的能力有限。一來由於瀏覽器大戰使得 HTML 規格快速被各家瀏覽器公司自行擴充，以致於現今的主要二家公司所發展的瀏覽器，對相同的 HTML 文件有不同的解讀結果，造成搜尋的困難。再者，由於 HTML 的上層資訊主要記載在 TITLE 及 META 二個標示之中，如果要搜尋 HTML 文件的內容，雖然只是讀取這二個標示的內容，但卻必須將整個文件下載，造成搜尋的工作變得相當沒有效率。基於上述的理由，所以 XML 被認為是定義上層資訊語法的最佳載具，因為它具有延展性、彈性以及可讀性。Web 瀏覽器使用的文件展示規則 XSL(eXtensible StyleSheet Language)，以及行動通訊所應用的無線標示語言 WML(Wireless Markup Language)，皆是 XML 在上層資訊的一種應用。基於 XML 於上層資訊表達的強大特性，使得將來各類型的應用程式將採用 XML 作為資訊內容表示。
2. 使用 XML 發佈及交換資料庫的上層資料庫(meta-database)：大部分以 Web 為基礎的三層式架構的應用程式，皆是透過 Web 伺服器向後端伺服器獲取

資料。但是透過 Web 伺服器所傳回用戶端瀏覽器的內容是由固定標示組成的 HTML 文件。這個文件對使用者來說只能當最終顯示結果，無法作進一步的應用。因此，建立一個介面使用 XML 可以透過此介面連結資料庫，並將資料庫查詢結果包含在 XML 自訂的標示之中。由於用戶可以事先知悉自訂標示的語意，所以將能作進一步的應用。

3. 使用 XML 當作應用程式之間訊息交換的上層傳輸協定(meta-protocol):XML 最強勢的應用在訊息交換,利用跨組織之間以及組織內不同應用程式之間的訊息交換。各組織利用 XML 可以進行應用程式與應用程式的訊息交換,利用協同合作的模式使不同的系統進行整合而發揮特定的綜效。由於各個企業有各自的資訊系統,其能接納的訊息是由企業自行定義,這之間如何進行整合,是經營企業間電子商務所必然面臨的問題與挑戰。傳統企業利用 EDI 標準來完成企業之間的商業文件交換,但是由於 EDI 建構在昂貴的專屬網路上,加以 EDI 標準沒有顧及企業個自不同的商業流程,應用 EDI 的企業就必須更改原有的流程規範,EDI 的應用變得相當沒有彈性。反之,由於 XML 彈性的語意表達,加上其為運作在低價的網際網路上的標準,所以克服傳統 EDI 昂貴且無彈性的缺點,使得 XML 文件迅速成為組織之間文件的交換標準。同樣的,未來政府單位及非營利性機關,亦可以透過 XML 制定符合公文規範的 XML 文件,來促進公文流程的效率及降低公文傳輸的成本。另一方面,利用 XML 描述的訊息文件亦能使組織內不同系統之間,建立彼此溝通的橋樑。使得不同應用程式可以進行整合而達到某種特別的應用。比如企業內的供應鍊管理系統可以與企業資源規劃系統,透過 XML 來進行訊息的溝通及系統的整合,使其能夠搭配完成電子商務系統的後端整合目的。由於 XML 可作為系統訊息交換的格式內容,使應用程式之間得以整合。所以,本論文引進 XML 架構進行研究,以期達成電子交易市集與參與廠商系統整合的目的。關於 XML 於系統整合上的訊息交換應用,將在本章

第二節中作更詳細的描述。

4.2 以 XML 為基礎的訊息交換平台

現在有許多應用程式皆是建構在現有的應用程式之上，目的在使現有資訊系統可以進行整合，而達成另一個特定的應用。例如電子商務的應用，就是將企業原有的企業資源規劃系統、供應鍊管理系統亦或是客戶關係管理系統等進行整合，使其能夠提供電子商務應用程式所需的原始資料或統計資料。由於不同的應用程式是在不同的主機或平台上執行，所以必須構建一套共通的語言或是在應用程式前端建立一個轉換程式。由於不同應用程式所需要的轉換程式不同，所以應用上不具彈性，因此建立共通語言的標準成為應用程式互通的主要方法。

4.2.1 傳輸協定與訊息格式

想要在現有應用程式之外建構一個新的應用程式，並使新的應用程式可以與現有的應用程式溝通，則建立一致的語言標準，是最有效且有彈性的作法。由此標準來描述應用程式之間傳輸的訊息，相關的應用程式必須共同遵從一致的傳輸協定及訊息格式標準，應用程式才得以整合，二者皆建構在網路應用層上：

1. 傳輸協定：傳輸協定是二個設備為傳輸資料而彼此認同的格式，在這是規範傳輸訊息的資料封裝方法以及傳輸管理。傳輸協定分成非同步傳輸協定及同步傳輸協定，非同步傳輸協定包含：電子郵件使用的 SMTP (RFC-821)、IBM 的 MQSeries 等。而同步傳輸協定包含：提供其他應用協定介面的 Sockets、Web 應用的 HTTP(RFC-2068)、CORBA/IIOP 使用的 IIOP 等。
2. 訊息格式：訊息格式定義訊息的語法，在一個既有的傳輸協定之上進行傳

送。訊息格式分成文字格式及二進位格式，文字格式包含：企業間訊息交換的電子資料交換(EDI)格式、SGML 標示語言等。而二進位格式包含：X.509 數位證書以及簡易網管協定使用的 ANSI.1 (Abstract Syntax Notation 1)、CORBA/IIOP 使用的 IDL(Interface Definition Language)等。

在建立應用程式的溝通機制上，發展最成熟的當屬 CORBA Common Object Request Broker Architecture)架構[11]。當初 OMG (Object Management Group)這個世界最大的軟體協會，成立的宗旨是建立一個物件導向的互通標準，使得分散式應用程式的開發可以透過這個標準建立可互通的分散式元件。目的亦包含使位處在異質網路環境中的物件導向軟體元件，具備再用性(reusability)、可攜性(portability)、互通性(interoperability)等特性。最後，OMG 期望在商務領域中定義相關的物件，在其技術層次上採用互通介面及協定，並定義物件管理架構(Object Management Architecture, OMA)，參考圖 4.1。OMA 在提供一個發展分散式物件標準的參考模型，這個模型定義分散式物件系統的關鍵元素。而 CORBA 標準則是定義在 OMA 架構中幾個關鍵的部分，而最重要的部分當屬 OMG 介面定義語言(Interface Definition Language)，以下簡稱 IDL。IDL 是敘述分散式物件的語言，負責描述可以實作獨立的物件介面。這也是為何 CORBA 架構能夠具備互通性的原因。當然介面定義如果沒有具體的物件實作將是沒有用的，基於這點 OMG 定義實作語言對應程式，使得開發人員可以從 IDL 的介面規格建立真實的物件。而語言對應程式是描述從 IDL 定義的型態轉換成原生語言的結構，如此的轉換使得程式設計人員可以運用其熟悉的程式語言，而不是只為了建立與其他應用程式溝通的機制而學習一套新的語言。因為目前的主流程式語言(如 Java)皆具備 IDL 對應程式作為其應用程式介面，因而可以利用該程式語言編寫應用於 CORBA 架構的 IDL。使用 IDL 使得分散式物件不論用什麼程式語言開發，只要分散式物件是建構在 CORBA 架構上，皆能利用 IDL 作為訊息傳送的内容，使不同程式語言開發的應用程式整合，所以 IDL 是扮演資訊系統整合中，訊息格式

的角色。

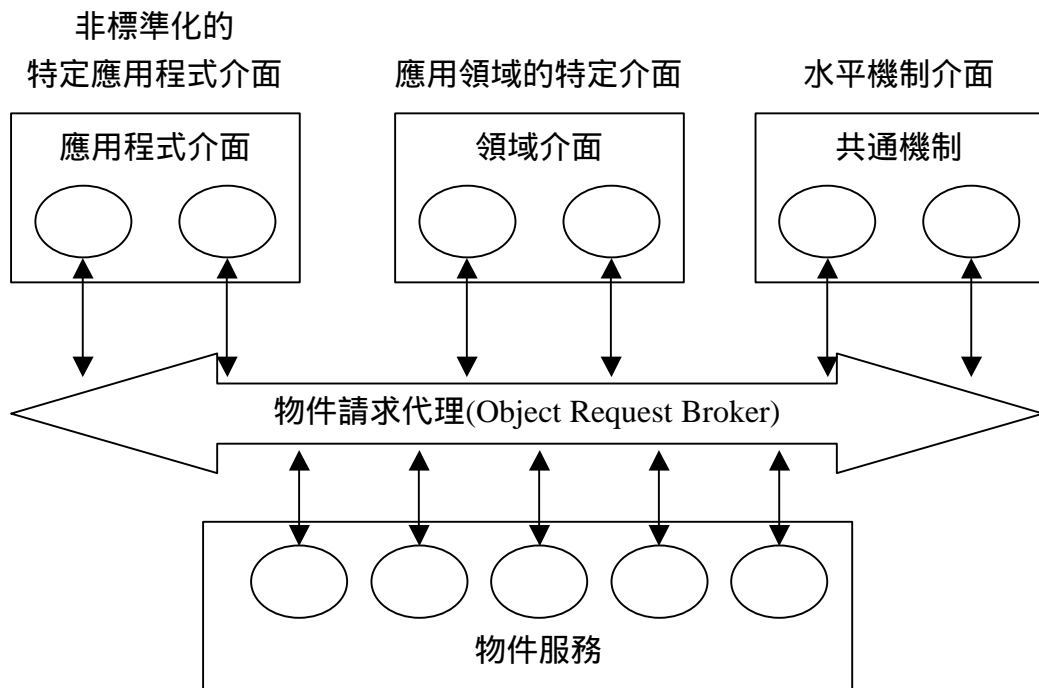


圖 4.1 OMA 參考模型[22]

在 CORBA 架構中，除了上述的 IDL 是系統整合必備的訊息格式之外。另一個需要討論的標的是系統整合所需的傳輸協定。在介紹 IIOP 之前，參考圖 4.1 可以發現，物件請求代理(Object Request Broker)，以下簡稱 ORB，扮演著架構中極重要的角色。ORB 可以視為 OMA 的基礎，它負責處理一個分散式物件系統中物件之間的通訊。ORB 接受用戶端的請求，找出並執行目標物件，然後轉傳遞用戶端的請求給該目標物件。ORB 提供基礎架構使得其他的元件可以一起運作，達到整合分散式物件系統的目的。在一個區域同質的網路中，只需要一個 ORB 即能實現 CORBA 架構，而元件的溝通是透過 ORB 的內部協定。但是在異質的網際網路環境中，必須提供 ORB 之間可以彼此通訊的機制，才能使 CORBA 架構適用於網際網路。所以 OMG 發展 IIOP(Internet Inter-ORB Protocol)，是支援 ORB 在網際網路彼此互通的傳輸協定，參考圖 4.2。所以 CORBA 架構在應用程式的整合，採用 IIOP 作為一致傳輸協定，以 IDL 作為一致的訊息格式標準。它的優勢在於使得個不同程式語言開發的應用程式，可以建立互通的架構而進行整

合。雖然前章曾述 Java 是跨平台的語言，即適用在網際網路異質的環境中。但是 Java 本身無法與其他程式語言開發的系統進行直接整合，因而採用 CORBA 架構成為一項選擇，而 Java IDL 也提供了實作 CORBA 架構的介面，足見 CORBA 在應用程式整合的重要性。

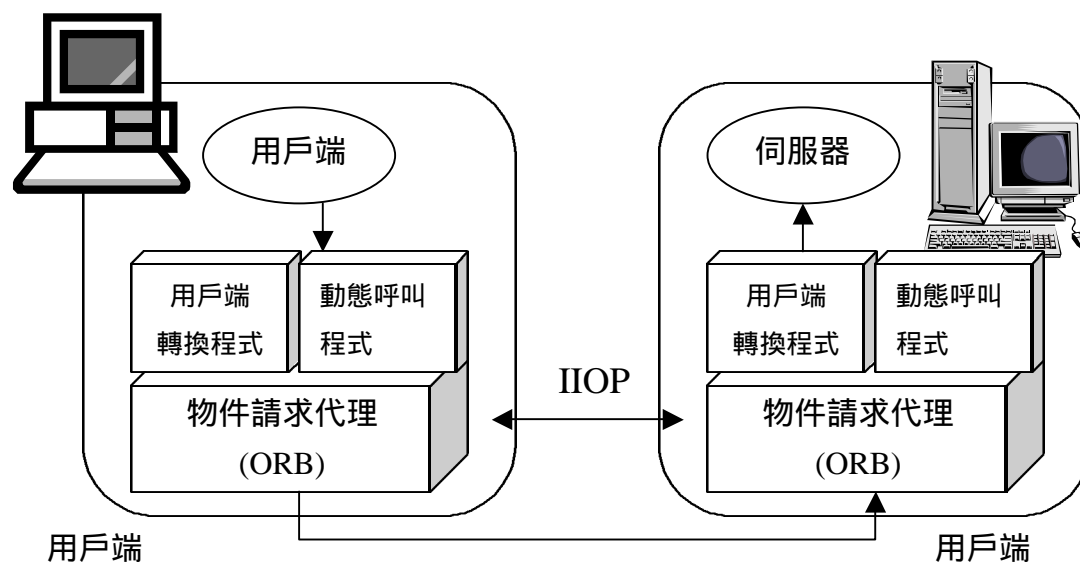


圖 4.2 CORBA 架構[11]

4.2.2 以 XML 為基礎的應用程式整合架構

前述提及 CORBA 架構使得應用程式得以互通整合。但是我們瞭解此架構的細節之後會發現 CORBA 是一個龐大且複雜的架構，由 CORBA 最新第三版的標準厚達千頁的情況來看，開發 CORBA 架構所需要的學習曲線將是採用它最大的阻礙。反之，XML 由於簡明易懂，而且從大家熟知的 HTML 語言為出發，使得學習及開發 XML，變成是一個不用花費太多時間即能進入狀況的應用程式整合方法。

Giacomo 等學者[12]提出四層式架構，來表示 XML 在網際網路應用程式的整合的方法，參考圖 4.3。這四個層由底層至上層分別為資訊層、中介層、介面

層、應用層。其中資訊層我們考慮資料儲存格式、儲存的媒體(如檔案、資料庫等)。中介層是將存在資訊層內不同格式的資料全部轉換成 XML 格式，將之儲存成 XML 文件或是動態產生 XML 結構。介面層負責公佈由不同應用程式元件或是通訊協定可以接受的訊息格式，這必須將中介層的 XML 格式轉換成特定的應用程式介面才能完成，本層的介面包含 XML 介面、CORBA 介面、XML 伺服器等。應用層則是各型應用程式的集合，包含 XML 瀏覽器、行動代理人、分散式物件等。應用層則是各型應用程式的集合，包含 XML 瀏覽器、行動代理人、分散式物件等。

在應用層上，利用 XML 描述的訊息格式標準(如 WML、XSL)的文件，透過任何一種傳輸協定，或是同由 XML 描述的傳輸協定(如 SOAP)，進行應用程式間訊息交換，以達到應用程式整合目的。即使以文字訊息交換為基礎的 XML 因為是以文字型態儲存，所以比以二進位機器碼型態儲存訊息的 CORBA 所需的容量大，因而前者較後者執行績效差。不過 XML 是人類及電腦皆懂的語言，加上 XML 文件包含上層資訊來表達傳輸的訊息，所以較能清楚表達文件的語意。所以採用 XML 架構比起 CORBA 架構來說，其最大的利基即在開發成本低而時效高，這也是 XML 在系統整合應用上比起 CORBA 迅速成長的主要原因。

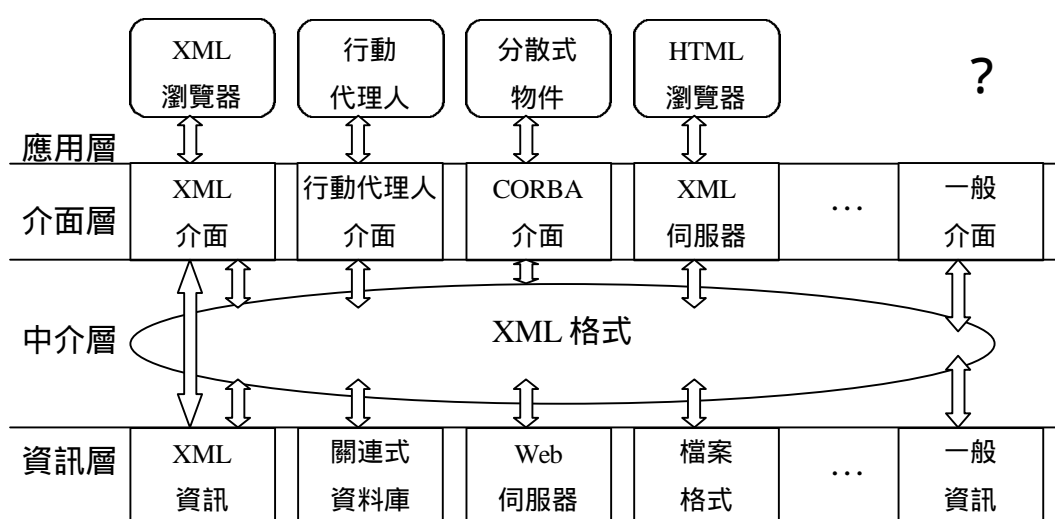


圖 4.3 XML 整合架構[12]

在電子商務的應用上 XML 的重要性尤其顯著，利用 HTTP 作為傳輸協定並以 XML 作為訊息格式，二者結合可以發揮簡易且彈性的效果。所以在企業進行電子商務應用時，在 HTTP 傳輸協定上使用 XML 訊息格式，成為主流的電子商務應用程式架構平台。如大眾所熟知的，網際網路的迅速發展起因於 Web 平台被用戶所接受，所以 HTTP 傳輸協定成為資訊傳送封裝的方法。由於市場需求使然，所以熟悉 Web 平台的程式開發人員比起熟悉傳統平台的人還要多出許多。而 XML 的文件格式與 Web 平台上的 HTML 文件格式相近，再加上配合以 HTTP 傳輸協定，使得 HTTP/XML 為基礎的訊息交換架構，將成為今後開發應用程式的主要參考模式。

另外，在本章 1.2 節言道，XML 應用在上層內容(meta-content)、上層資料庫(meta-database)以及上層傳輸協定(meta-protocol)三個領域中，這意味三個領域皆會出現以 XML 描述的語言。應用在這三個領域中的 XML 語言，由於 XML 本身為自我描述的語言或謂上層語言(meta-language)，因此我們將它們分別謂之 XML 內容語言、XML 資料庫語言、XML 傳輸協定語言。前面曾言，應用程式間欲進行整合，必須在應用層上遵循一致的傳輸協定及訊息格式標準。結合這個概念，配合 XML 的應用範圍，筆者設計一個以 XML 為基礎的應用程式整合架構，以圖 4.4 表示。

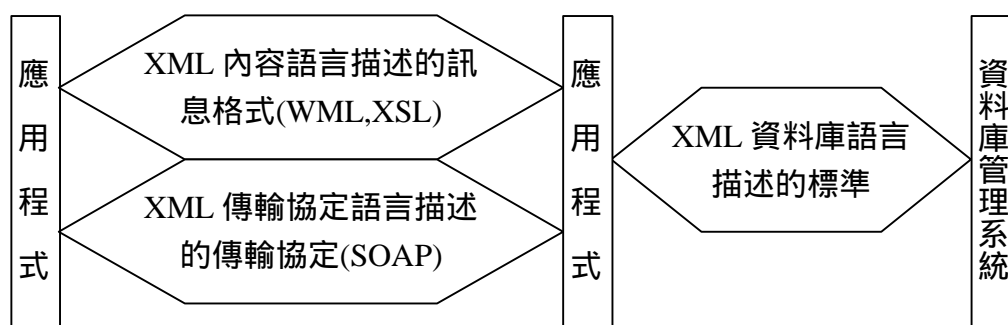


圖 4.4 XML 應用範圍與應用程式整合架構

4.2.3 支援應用程式整合的 XML 協定

為了要讓不同的程式語言開發的應用程式可以進行整合，那麼必須要讓程式語言具備存取 XML 文件的能力。如同 CORBA 架構使用 IDL 作為各種程式語言實作 CORBA 的介面一樣，XML 架構使用所謂文件物件模型(Document Object Model)，以下簡稱作 DOM，來讓各種程式語言建立實作 XML 的介面。DOM 簡單來說是有效的 HTML 文件以及合法的 XML 文件的應用程式介面[1]，它定義 XML 文件的邏輯結構以及存取此文件的方法。利用 DOM 程式開發人員可以建立 XML 文件，並進行後續的維護。任何存在 HTML 或 XML 文件的元素都可以透過 DOM 來進行存取。也就是說，將文件視為一種文件物件，而透過提供此物件介面的程式語言來對此物件進行存取。設計發展 DOM 的 W3C 期望 DOM 可以被任何程式語言所支援，而為達到此目標，所以 W3C 選擇在 OMG IDL 上定義 DOM 介面規格。因為 OMG IDL 設計的目的即在促使不同程式語言發展的程式可以整合，所以在 OMG IDL 上發展 DOM 介面規格，將有助於將 CORBA 架構亦納入 XML 整合架構中。再者，由於 Java 是開發網際網路應用程式的主流程式語言，因此 W3C 也定義支援 Java 的 DOM 介面規格。另外，在 Web 瀏覽器常用的 JavaScript 也有 DOM 介面來支援其存取 XML 文件，例如 Microsoft IE 5.0 瀏覽器即支援 DOM，我們可以用 Web 開發人員熟悉的 JavaScript 透過 DOM 介面，在瀏覽器執行環境中對 XML 文件進行存取。DOM 標準的建立在使各型應用程式可以透過實作它而建立及維護 XML 文件，進而使各應用程式具備利用 XML 進行訊息交換，達成系統整合目的。

另外，W3C 在 XML 在應用程式整合的應用上，特別發展了一套命名為 SOAP (Simple Object Access Protocol)的協定[2]，它是一個透過 XML 作為描述標準，在分散式運算架構環境下，提供一個應用層上的訊息傳輸協定。上述的 DOM 是提

供各式程式語言存取 XML 文件的能力，重點不在訊息交換本身，而 SOAP 則是專注在應用程式整合領域上，作為 XML 文件交換的橋樑。SOAP 是一個用在分散式環境中資訊交換的協定，它是建立在 XML 為基礎的協定，目前 SOAP 標準在考量如何整合在 HTTP 之中，未來將會將 SOAP 擴充整合在其他的傳輸協定。SOAP 存在三個主要部分：第一個部分是 SOAP 信封，定義訊息內容的結構，誰應該處理它，無論它是選擇性或是必要性的。使用 SOAP 信封描述的 SOAP 訊息結構，以圖 4.5 表示。第二部分是 SOAP 編碼規則，不同應用程式皆有它各自的資料型態定義，所以 SOAP 編碼規則提供一個序列化機制，使得不同應用程式使用的資料變數可以進行交換。第三部分是 SOAP 遠端程序呼叫(RPC)，利用 XML 文件交換達成應用程式遠端程序呼叫的目的。在 SOAP 發展更成熟並加入安全機制之後，運用各自專屬協定的應用程式將可以透過 SOAP 進行整合。

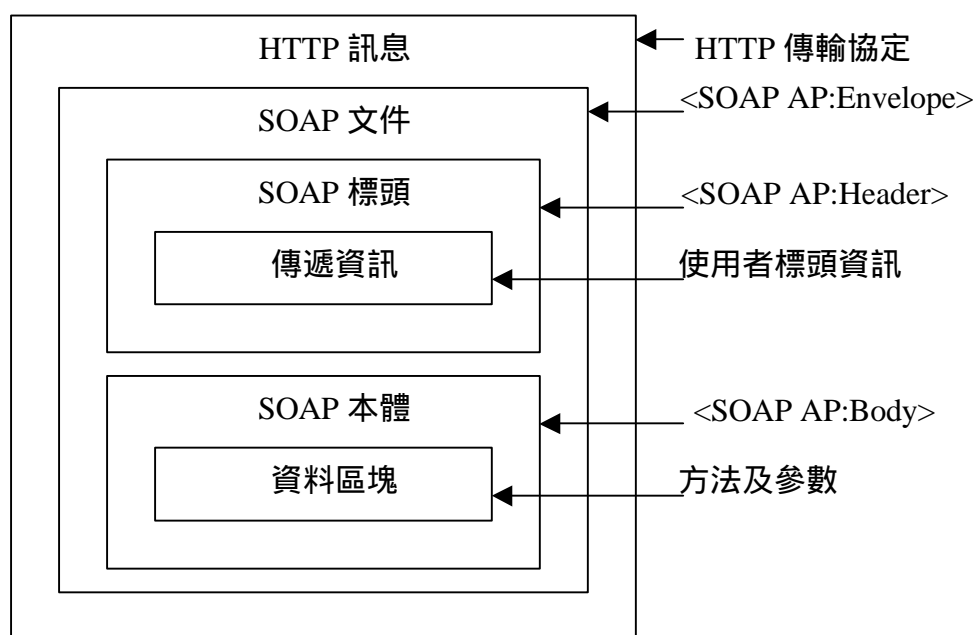


圖 4.5 SOAP 信封描述的 SOAP 訊息結構[2]

4.3 電子交易市集相關的 XML 商務標準

前述曾提及企業對企業電子商務擁有廣大的市場潛力，目前業界在電子交易市集的著力點即在企業對企業電子商務，而狹義的電子交易市集經營的是公眾式開放市場，市集是由第三者主導，並有多個顧客及多個供應商參與。筆者認為一個電子交易市集要能夠吸引企業加入，必須符合三個條件：

1. 參與與退出的彈性，對鼓勵企業參與電子交易市集是相當重要的。為求顧客及供應商可以彈性參與，則電子交易市集的系統架構必須與顧客及供應商的系統易於相容。
2. 電子交易市集本身提供顧客及供應商資訊傳輸的開道，使得買賣雙方因為資訊語意表達可以互通而順利進行交易。
3. 不同電子交易市集之間的資訊傳輸，更是電子交易市集不可或缺的功能。在參與市集的供應商或是顧客無法尋找合理交易條件時，那麼將此類資訊轉送至其他電子交易市集，以提升服務品質，對電子交易市集能否成功是相當重要的。

上述三點最有效的解決方法就是遵循企業對企業電子商務的標準。如今電子交易市集最普遍的地方在電子商務發展最成熟的美國，在電子交易市集的領域中，最被廣為接受當屬可以自我描述的 XML 語言，它的核心價值在於資訊系統的整合，關於這個語言的內容在前面已經作詳盡介紹。但是如果不同的供應商、顧客以及電子交易市集，自行利用 XML 語言來描述其商務相關的標準，那麼 XML 對企業對企業電子商務的價值將大打折扣，無法達成其資訊系統整合的目的。

的。為此，介紹幾個支援企業對企業電子商務的 XML 標準，推動這些標準的機構包含 OBI、eCO、RosettaNet、commerce XML 及 BizTalk 等最具代表性。這五個機構以 XML 為基礎推動可以運用在企業對企業電子商務的標準，非常有助於電子交易市集的推展。這些機構及其推動的標準分述如下：

1. OBI

OBI(Open Buying on the Internet)建立一套企業對企業電子商務交易的標準，交易的標的物是大量且低價的產品。參與 OBI 的電子交易市集可以利用數位憑證及電子簽章完成鑑別及存放顧客產品清單的機制，建立了可信賴的安全規範來管理電子交易市集。OBI 的特色在於企業可以容易地利用動態互通的交易網站，進入此市集成為交易會員。OBI 聚集了企業的買方市場，吸引賣方加入此市場以提供買方所需貨品，所以 OBI 是利用買方主導市場的趨力來建構其交易市集。不像傳統由賣方企業來維護買方資訊，OBI 的交易市集將責任二分化給買方及賣方單位，買方單位負責維護需求品資料、帳單代碼、稅率狀態及合約；賣方單位則負責維護顧客目錄及內部工作流機制。OBI 計畫是在補足傳統 EDI 之不足，而非取代它。由於 OBI 強調其簡易的交易環境，所以它輕易地整合企業的採購應用系統而獲致低的維護成本。目前 OBI 建立的標準是第二版，是以 CGI 及 HTTP 為基礎，未來的版本將採用 XML 訊息交換標準。

2. eCo

eCo[24]是由 CommerceNet 在 1996 年提出及創建的系統，CommerceNet 是一個由超過三十五家企業代表所組成的協會，主要由 Commerce One 這家公司贊助。在 1997 年，由於 XML 的簡易性加以其被 IBM、Microsoft、Netscape 及 Sun 等資訊大廠所廣為採用，所以該協會決定將原先以 CORBA 為基礎建立的互通架

構，換成以 XML 標準作為其建構基礎。eCo 對企業對企業電子商務常重要的貢獻，在於各公司可以容易地透過 eCo 系統建立的規格就能與其現有的供應鍊結合，找到該供應鍊上的所有潛在顧客及供應商。由於發展規格的業者亦包含原先就從事此領域的組織，如 OBI 及 RosettaNet，所以可想見其將來對企業對企業電子商務的重要性。以 XML 為交換基礎的 eCo 系統，可以允許各企業之間進行自訂的 XML 文件通訊，由於文件是企業內部自行定義的，所以除了可以與其內部資訊系統整合之外，更可以透過 eCo 的代理系統，讓交易夥伴可以瞭解此文件。刊載在 eCo 網站上的商業介面定義(Business Interface Definition)是讓交易雙方可以互通交易文件的關鍵，它會告知潛在交易夥伴該公司提供的服務以及如果選擇此服務後應用要用什麼文件。例如，商業介面定義可以讓顧客提出採購訂單訂購商品，亦可讓供應商透過貨品清單狀況報告的下載來檢查其可供應商品(圖 4.6)。

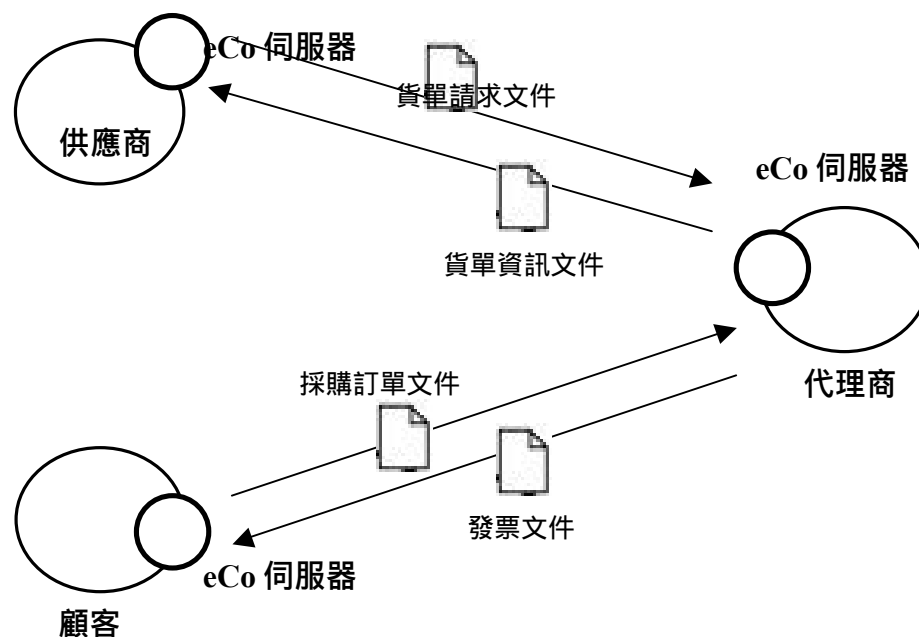


圖 4.6 以 XML 為文件交換基礎的 eCo 系統[24]

3. RosettaNet

RosettaNet 協會在資訊科技及電子元件產業上建立以 XML 為基礎的供應鍊

管理商業標準。這個協會為從事電腦軟硬體的高科技企業發展互通的電子商務標準，它定義商業程序並提供資料交換的技術標準。Rosettanet 使用商務夥伴介面程序(Partner Interface Processes, PIPs)來訂定企業與其商務夥伴的溝通介面。為了開發 PIPs，此協會定義二個其他的元件：參考字典及實作架構。參考字典包含了技術及商業目錄以定義訊息的詞彙。其中的技術目錄是確保企業使用相同的語言，並且在進行電腦元件及週邊設備產品資訊通訊時，可以了解相同的特徵值。而商業目錄是定義商品價目表、貨品交易、交易對方企業的商業屬性(付款、訂單、運送等)的語言。實作架構在定義安全訊息交換的協定，包含訊息格式、訊息內容、網路架構及安全機制。

除了上述的參考字典及實作架構之外，RosettaNet 定義組織的獨立商業程序模式。在這個模式中使用訂單管理員及商品清單公佈員的管理程式，以明確定義商務夥伴的職務。將參考字典、實作架構及商業程序模式輸入商務夥伴介面程序之後，RosettaNet 將可以公佈此程序至其交易夥伴，而收到此程序的公司將能夠將此資訊作為指引而發展他們的軟體。透過 RosettaNet，在供應鍊上的不同二個組織將可以標準化他們的介面並擴展此介面在現有的架構上。

4. commerce XML

由 Ariba 為代表的超過四十家企業支持的 commerce XML，簡稱 cXML，是一個公開的網際網路標準，設計目的在利用少量的 XML DTD 集合來達到貨品清單內容及交易資訊能夠彼此交換。cXML 規格訂定二個訊息協定模式：請求 / 回應模式(request/response model)及非同步單向模式(asynchronous one-way model)。前者是利用 HTTP 協定為基礎，以同步化的作業方式處理用戶端與伺服器之間的回應及請求訊息；後者是客戶端將對伺服器的請求加以編碼，然後傳送至伺服器進行處理，由於伺服器不會同步給予用戶端回應，所以稱為非同步單向模式。

cXML 交換的訊息包含標頭及資料二個部分，其中標頭部分記錄鑑別資訊以提供其應用的安全機制。cXML 的規格專注在非生產性元件上，而其提供的資訊服務內容包含了運送通知、狀態通知、確認訊息及付款手續等。cXML 最重要的優勢在簡易的使用及實作。

5. BizTalk

BizTalk 的入口網站及以 XML 為基礎的技術規格，促使傳統的資訊系統得以整合並提升企業的電子商務效率。利用 XML 的標籤、元件及屬性來開發訊息交換的系統，並將企業的 XML 文件在此入口網站上進行合法化、敘述、註冊、儲存。BizTalk 著注在提供 XML 及非 XML 的資料，並將傳統以 EDI 為資訊交換基礎的傳統系統可以進行轉換，以提升其互通性的價值。

以上是目前在企業對企業電子商務的相關標準推動領域中，最具影響力的組織及其發展的標準。這五個主要的領域依據其目標產業、安全性、通訊協定、訊息格式、查詢機制、擴充性以及交換標準等特徵加以比較，可以用表 4.1 來表示。

特徵	OBI	eCo	RosettaNet	cXML	BizTalk
目標產業	非生產性元件	未指定	資訊電腦零件	非生產性元件	未指定
安全	SSL 及數位憑證	選擇性	SSL,數位感證,數位簽章	訊息標頭包含的鑑別資訊	現存的標準
通訊協定	HTTP	HTTP	HTTP/CGI	HTTP,URL 編碼表單	HTTP/MSMQ
訊息格式	EDI 文件	XML 文件	合法的 XML 文件	XML 文件	自訂標籤的 XML 文件
擴充性	適用於一對一交易	充足	實作指標可以擴充	利用 XML DTD 加以擴充	中央式儲存造成處理受限
交換標準	EDI X12 資料目錄	共用企業函式庫	技術及商業字典	XML 標籤	自訂標籤

表 4.1 XML 商務標準比較[28]

第五章 型錄搜尋代理人架構需求與架構方案研究

5.1 架構需求分析

在電子交易市集的應用環境中，茲提出一個型錄搜尋代理人的構想。目前的電子交易市集多以提供供應商型錄為利基市場，避免涉及顧客與供應商的交易行為，以獲得用戶的青睞。這類市集提供的型錄，可以歸類為產品規格及功能詳述、電子郵件詢問以及即時的商品交易資訊型式。從提供顧客供應商型錄仲介的觀點出發，可以設計一個加值的型錄資訊服務，茲將這個型錄資訊服務稱作型錄搜尋代理人，它主要的功能在提供即時商品交易資訊，包含可供數量以及即時價格，以協助顧客作交易決策。在第三章曾言，軟體代理人依其特性可以分成數種型式的代理人，而這個所謂應用在電子交易市集的型錄搜尋代理人，其目的在採集供應商現有的電子型錄內容，並將之綜合整理成一個市集中型錄資訊。透過網際網路搜尋目標資源，再將此資源再利用而成為市集的型錄資訊，因此它強調可以幫助瀏覽器用戶更容易清楚的掌握型錄相關資訊。所以我們將這個型錄搜尋代理人定位為：具備行動代理人能力的資訊代理人，亦即它是兼具行動代理人及資訊代理人特性的複合式軟體代理人。從顧客的觀點來說，電子交易市集中的型錄搜尋代理人，代理顧客蒐集供應商的型錄資訊，經由條件過濾及型錄彙集的過程，提供顧客加值的型錄服務。電子交易市集的價值即在其如何提供增值服務，由於電子交易市集本身既不是產品供給者，更不是產品需求者，它的存在性即在它如何提供增值服務，使得供應商及顧客都少不了它。而型錄搜尋代理人的貢獻，即在協助電子交易市集提供增值服務。

Lee, Whang 二人[15]曾經提出一個資訊交換中心(Information Hub)的理念。資訊交換中心是多個組織為求供應鍊整合，而彼此透過網路交換資訊的地點。各

個參與者內部的應用程式(包含 ERP 或是其他企業系統), 透過這個資訊交換中心可以與其他企業的應用程式傳達彼此的資訊, 建立溝通的管道, 茲以圖 5.1 示意。在電子採購的應用中, 電子交易市集平台上運作一個資訊交換中心, 可以彙集多個供應商的商品相關資訊, 提供連上電子交易市集平台進行採購的顧客, 可以得到完整的商務資訊。

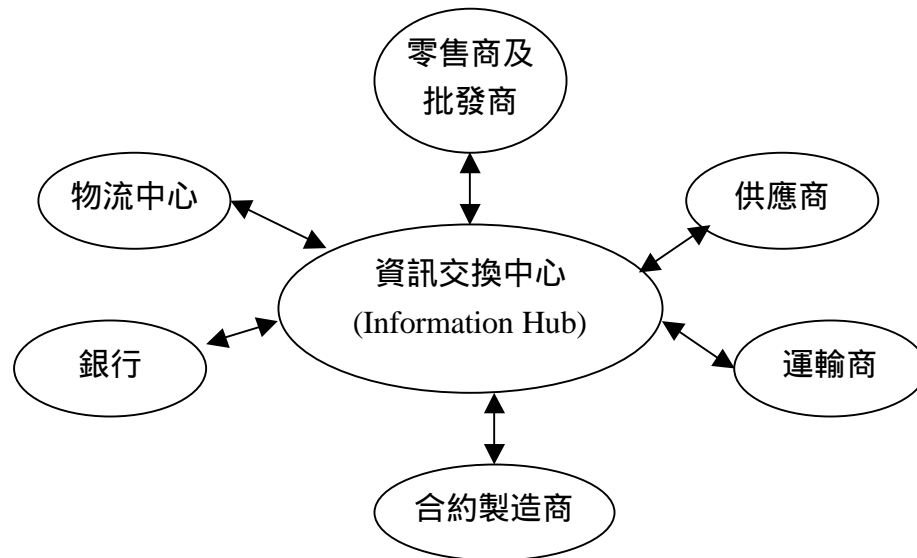


圖 5.1 資訊交換中心架構[15]

由於企業資訊委外, 以及強調企業協同合作提升競爭力的趨勢, 一個資訊交換中心的架構, 將成為未來資訊提供的主要模式。應用在電子交易市集中的型錄搜尋代理人即為一例。Web 是目前人類透過網路擷取資訊的主要介面, 所以電子商務的應用無法脫離 Web 平台提供的服務。IBM 的 Web 服務架構團隊曾經提出一個 Web 服務的概觀: Web 服務架構要能夠真正實用, 必須有三個角色: 服務提供者、服務仲介者、服務請求者, 服務請求者透過服務仲介者搜尋到服務提供者, 三者的關係以圖 5.2 表示。這個觀念最普遍的應用是入口網站(如 Yahoo), 我們透過入口網站找到相關網址, 再連結網址而查詢相關資訊。這個觀念其實與前述的資訊交換中心架構理念相同, 因此將此概念稱為 Web 資訊交換服務架構。Web 資訊交換服務架構的目的在達成企業內部, 或是企業與企業之間的應用程式

整合，所以也可以稱為此架構為應用程式整合架構。其中服務仲介者平台上有一個負責處理服務請求者的請求並給予回應的軟體代理人，它首先決定服務提供者為誰，並轉述服務請求者的請求至服務提供者的平台上進行處理，等到接受到服務提供者的回應訊息後，它將之作條件過濾及彙集的程序。整個流程如圖 5.3 所示。

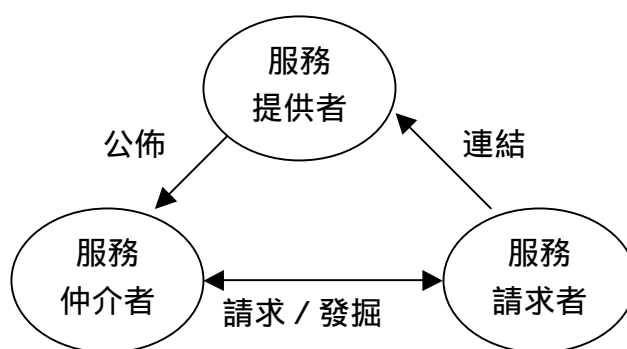


圖 5.2 Web 資訊交換服務架構

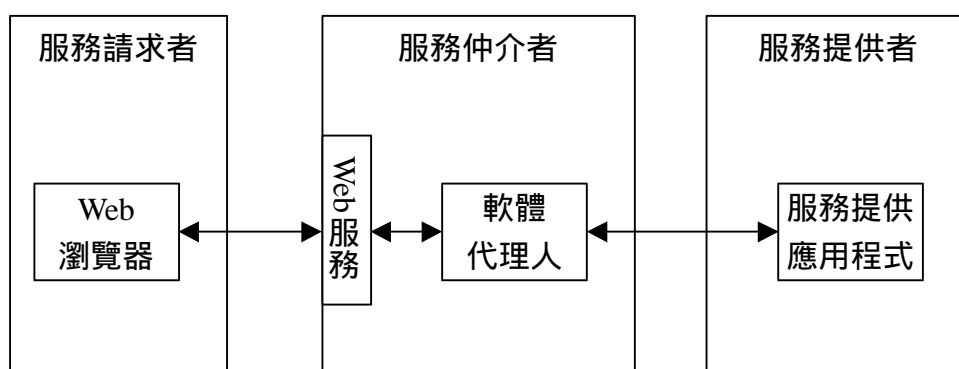


圖 5.3 Web 資訊交換服務架構

Web 資訊交換服務架構，是現今資訊委外下必然的產物。在旅遊服務(參考圖 5.4)以及投資諮詢服務(參考圖 5.5)的應用中，我們會發現提供這類服務的代理人，或謂經紀人，他們有足夠的理念需要實作這類架構。而電子交易市集是供應商與顧客的代理人，同樣需要 Web 資訊交換服務架構，來實現型錄搜尋代理人的理念(參考圖 5.6)。接下來二節，將對 Web 資訊交換服務架構的相關解決方案作進一步探討，以研究可供型錄搜尋代理人架構運作的參考模型。首先介紹行動

代理人架構方案，然後介紹三層式 XML 架構方案，雖然這二個方案所提的架構案例，不一定是在電子交易市集的應用，但卻皆以 Web 資訊交換服務架構為基礎。

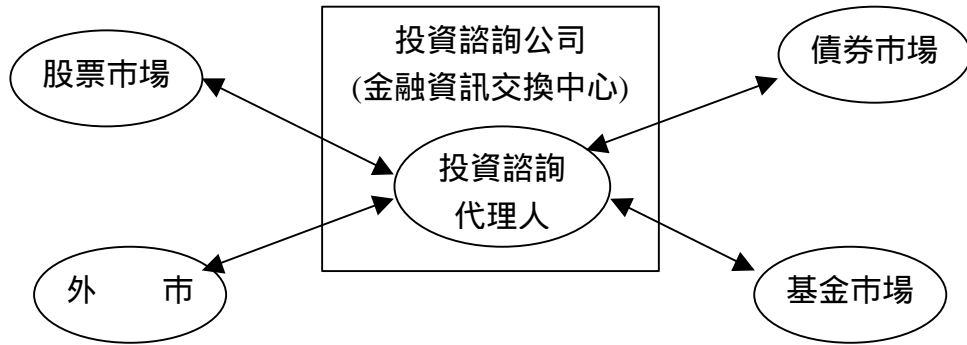


圖 5.4 投資諮詢應用的 Web 資訊交換服務架構

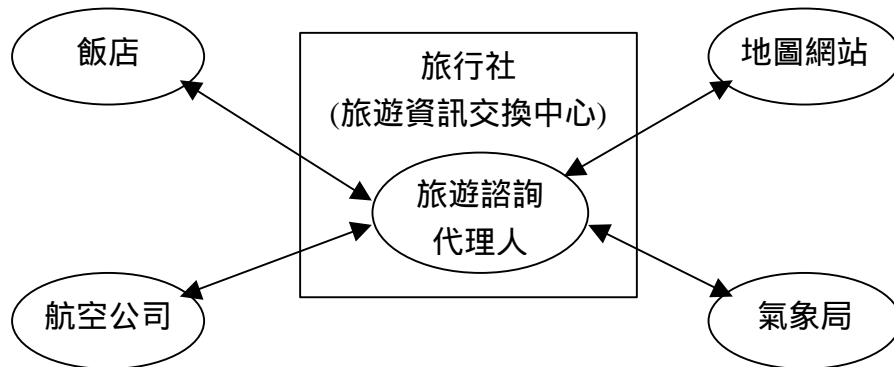


圖 5.5 旅遊諮詢應用的 Web 資訊交換服務架構

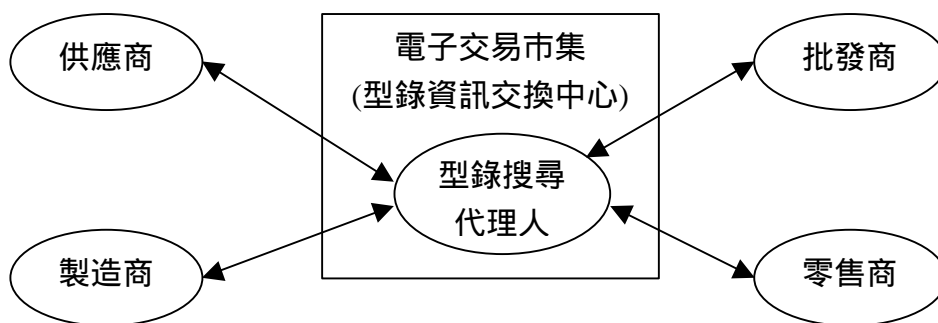


圖 5.6 型錄搜尋應用的 Web 資訊交換服務架構

5.2 行動代理人架構方案

雖然行動代理人架構未被普遍應用，但是利用行動代理人架構所建置的電子交易市集平台，卻有不少學者專家提出相關研究。這些架構包含實際運作的系統，以研究性質的雛型系統。TabiCan 使用 IBM Aglets 行動代理人系統開發市集平台，TabiCan 是一個提供航空訂票及旅行住宿服務的電子交易市集[6]。而 Sooho[27]等學者提出一個行動代理人架構，以 Web 資訊交換服務架構為基礎，應用在電子交易市集平台上，為一個研究性質的雛型系統，參考圖 5.7 所示。

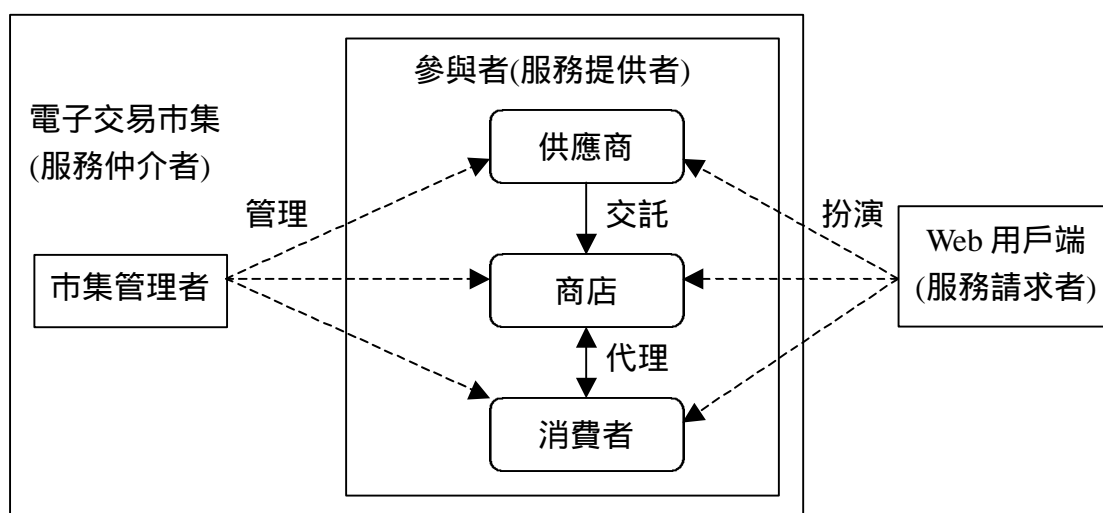


圖 5.7 Sooho 電子交易市集架構[27]

在這個架構中行動代理人可以幫助人們加入這個市集，並且可以滿足使用者不同的需求。市集是由管理者及參與者所組成，管理者提供市集運作環境而參與者參加這個市集的電子商務活動。管理者有二個主要工作，一個是建立參與者專用的代理人，一個是建立參與者共通的商品型錄。把這二個工作交給市集管理者可以減輕會員加入市集的學習障礙，並且可以保持產品資訊定義的一致性。而會員包含供應商、商店以及消費者，任何單位或個人可以扮演其中任一角色。管理

者第一個工作是建立及維護該市集的商品共通型錄，供此市集的應用系統使用，使得交易的標的產品可以透過此型錄而達到產品資訊交流。

管理者的另一個工作是維護會員專屬的代理人，管理員透過其管理員管理代理人產生並提供會員三種代理人，分別為設定代理人、會員管理代理人、型錄交託或查詢代理人。不管是供應商、商店以及消費者等市集的會員，安裝成為市集會員平台的步驟皆相同。皆是透過管理者的電子交易市集平台，利用 Web 瀏覽器上 Applet 介面的設定代理人，輸入參與者及型錄相關資訊。日後利用身份鑑別程序，參與者同樣透過設定代理人進行個別資訊的維護。會員管理代理人依據參與者角色不同可以分為供應商管理代理人、商店管理代理人、消費者管理代理人三種，其任務為維護存在參與者平台上的相關資料，包含商品型錄以及用戶個別資訊。型錄交託代理人是供應商將其商品型錄資訊傳播至各個相關商店的代理人，而型錄查詢代理人是消費者查詢商店的商品型錄資訊的代理人。型錄交託代理人的運作，如圖 5.8 所示。型錄查詢代理人的運作，如圖 5.9 所示。在圖 5.9 中，如果消費者連線則召回至消費者平台否則召回至電子交易市集，等待下次消費者連線再從電子交易市集平台召回。這個架構提供極佳的容錯能力，如果供應商依據會員資料庫派遣的產品交託代理人至相關的商店，那麼即使該商店主機發生問題，亦能將該代理人存在市集管理平台之中，等待商店主機正常才從市集管理平台派遣至商店平台。有效交派代理人替使用者完成交易的任務。

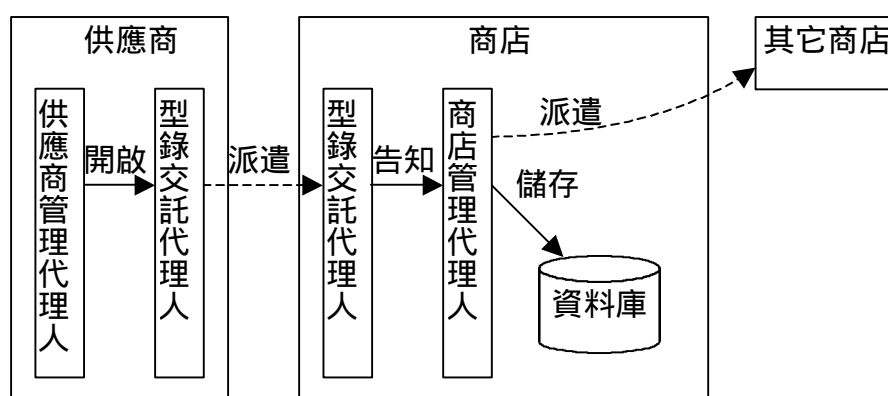


圖 5.8 供應商型錄交託程序[27]

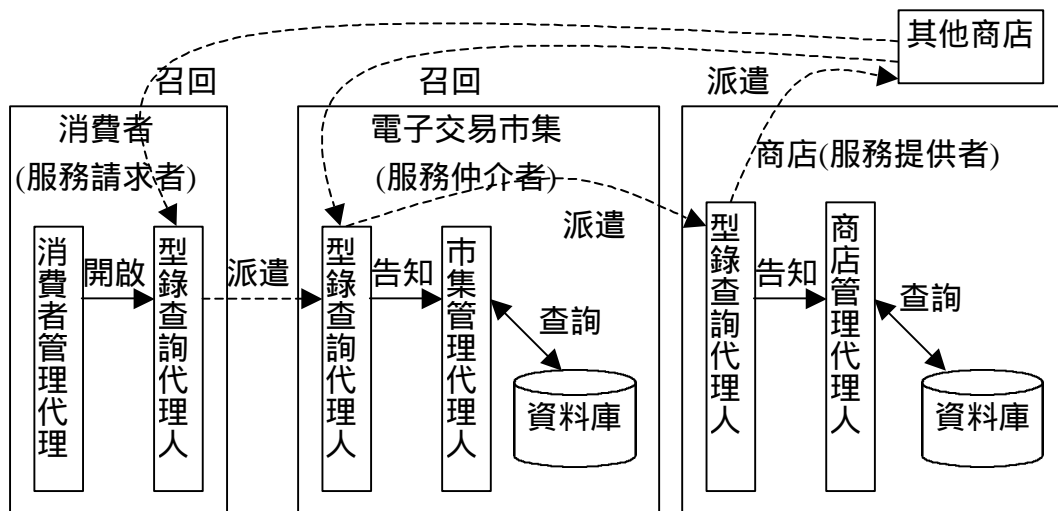


圖 5.9 消費者型錄查詢程序[27]

- 架構評論:

這個利用行動代理人架構建置的電子交易市集平台，利用 Java 開發的 IBM Aglets 行動代理人系統所建置。發揮 Java 架構的特性可以跨平台執行，使得市集參與者僅要安裝 Java 執行環境以及 IBM Aglets 行動代理人系統，即能透過電子交易市集平台的 WWW 網站，加入此電子交易市集。由於市集平台定義商品型錄格式一致的介面，因此對參與者來說，僅需要透過 Applet 介面即能夠管理型錄或是查詢其它參與者的型錄。然而定義一致介面隱含另一個重大的缺點，表示參與者管理型錄格式的權利被剝削。另外，如果參與者原有型錄相關的資訊系統，那麼電子交易市集提供的型錄資訊，需要再另行維護，造成人力資源上不小的負擔。因此，筆者的看法是如果電子交易市集的成員，大部分是個人或是小型企業，則利用這個架構管理商品型錄，將是比較能夠被接受的。

5.3 三層式 XML 架構方案

在上述行動代理人架構方案中，評論到該架構有個很重大的缺點，在於參與者需要再行維護額外的型錄資料庫，供電子交易市集平台搜尋使用。克服這個問題最佳的方法就是不要再另行導入額外的型錄資料庫，而是從電子交易市集的型錄搜尋應用程式，與供應商的型錄服務應用程式進行整合做起，此方法同樣可以達到型錄搜尋彙集的目的。應用程式整合最好的架構策略就是使用 XML 架構，以下提出三個 XML 結合架構皆是以三層式架構為基礎。

5.3.1 XML/HTTP/Servlet 架構

Maruyama, Kent Tamura, Naohiko Uramoto 等人在「XML and Java Developing Web Application」一書[13]，使用 Java 程式語言開發 Web 應用環境，其中並提出一個以上述所謂的 Web 資訊交換服務架構。這個架構使用 Servlet 作為 Web 服務開發環境，Servlet 提供一個開發 Web 伺服器程式的環境，本身具備接受用戶端 HTTP 請求，執行內部 Java 開發的程式並給予回應，是開發 Web 伺服器程式不可或缺的工具。該架構使用 XML 作為訊息交換的標準，以達成系統整合的目的。由於 XML 訊息交換是透過 HTTP 傳輸，一般稱此模式為 XML/HTTP 或是 XML over HTTP。

該架構之運作流程：首先服務仲介者的 Servlet 接受服務請求者送來的 HTTP 請求，然後 Servlet 呼叫軟體代理人。軟體代理人查詢服務提供者的網址，如果服務仲介者僅需要即刻回應，則軟體代理人只呼叫 HTTP 用戶端一次，如果服務仲介者設定在一段時間內回應，則軟體代理人啟始一個可以獨立執行的 Thread

程式，在設定的時限內固定頻率呼叫 HTTP 用戶端。然後 HTTP 用戶端執行在服務提供者平台上的 Web 服務，這個 Web 服務由參數來決定回應的結果，所以服務仲介者僅需要瞭解參數值的意義，即能向該 Web 服務提出一個 HTTP 請求。該 Web 服務執行此 HTTP 請求，存取後端的資料庫並且將回應的訊息以 XML 描述，待服務仲介者的軟體代理人接受此回應之後進行處理，然後將訊息傳回服務請求者。

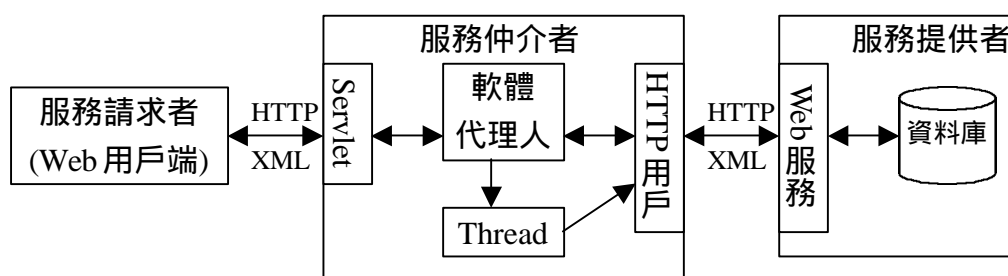


圖 5.10 XML/HTTP/Servlet 架構

- 架構評論：

這個架構透過目前應用最廣的 Web 服務來開發資訊服務環境，透過 Web 使用的 HTTP 協定傳輸 XML 作為訊息交換，以達成系統整合的目的。在同步回應模式中，服務仲介者僅需知道服務提供者的 Web 服務的網址及程式名稱和參數值，便能下載服務提供者提供的 XML 文件。這種模式將作為下一章架構設計時的參考模式。在非同步回應模式中，這個模式則相當有可議之處。非同步回應模式，該架構在服務仲介者平台上使用 Thread 監聽程式，固定頻率呼叫 HTTP 用戶端以進行服務提供者所提供的 XML 文件下載。這個方式由於需要將 XML 文件不斷下載處理，直至達成軟體代理人既定的目的或是設定的時限到達。如果說服務仲介者接受到服務請求者的要求之後，即開啟一個 Thread 去執行一個監聽任務，一旦服務的請求過多，將形成服務仲介者與服務提供者之間的網路極大的負荷，並且造成服務仲介者系統極大的負擔。因此接下來討論二個架構，可以在非同步回應模式中作出一個改良。

5.3.2 XML/SOAP/Servlet 架構

Growbal[34]利用 Java 開發一個 Web 資訊交換服務架構，這個架構在此稱作 XML/SOAP/Servlet。與上述架構最大的不同處在於此架構導入 SOAP 協定，SOAP 是一個傳輸 XML 文件的協定，本身支援 SMTP 以及 HTTP 傳輸協定，透過 SOAP 我們可以加強 XML over HTTP 的能力。上述的架構僅能在 HTTP 回應中傳回 XML 文件，無法在 HTTP 請求中傳輸 XML 文件。

該架構的運作流程：服務仲介者接受服務請求者的 HTTP 請求之後，透過 Servlet 介面啟動軟體代理人，軟體代理人查詢服務提供者的網址，並且將用戶的請求描述成 XML 文件，呼叫 SOAP 用戶端程式將此文件包裝在 SOAP 定義的文件之中，然後將之送至服務提供者的 Servlet 介面，這裡的 Servlet 接受到此 XML 文件之後，呼叫 SOAP 服務程式，並且將此 XML 文件送至 SOAP 服務程式進行解析。SOAP 服務程式經過 XML 解析，然後呼叫內部的應用程式，再將回應訊息以 XML 描述，並且送回服務仲介者，待服務仲介者的軟體代理人處理完此 XML 訊息之後，將之送回至服務請求者。關於 SOAP 在第四章有更詳盡的描述，該架構的運作流程如圖 5.11 所示。

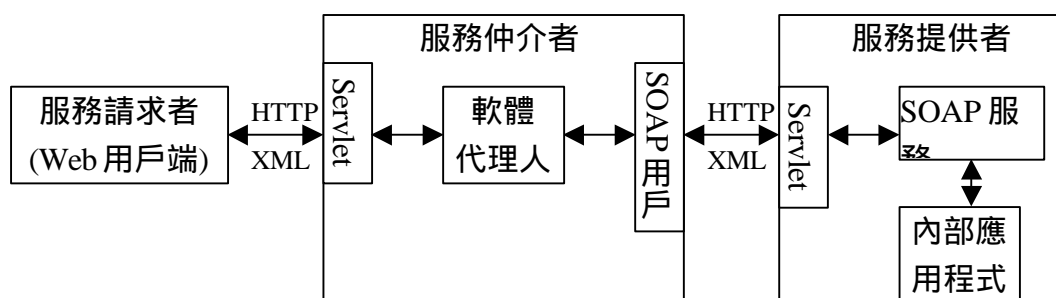


圖 5.11 XML/SOAP/Servlet 架構

- 架構評論：

SOAP 協定提供一個良好的 XML over HTTP 運作環境，透過 SOAP 用戶端與 SOAP 服務的協同運作，使得 XML 文件可以容易地進行傳輸與提取。與上個架構相較，其最大的改良在於服務仲介者不需要啟動一個 Thread 程式，來隨時監聽服務提供者的應用程式服務。因為擷取資料庫的任務移到 SOAP 服務程式來執行。因此服務仲介者僅需要透過 SOAP 將 XML 文件送至 SOAP 服務，即能達到與服務提供者內部應用程式整合的目的。僅需要一次的 XML 傳輸，即能達到監聽服務提供者的 Web 資料庫服務的目的。然而該架構有二個缺點極待改進，第一個缺點是該架構不論服務請求者提出同步或非同步回應的請求，皆透過 SOAP 來進行處理，所以比起 XML/HTTP/Servlet 架構中的同步回應模式還來得複雜且較無效率。第二個缺點是服務仲介者的 SOAP 用戶端，以及 SOAP 服務，必須配合服務提供者的內部應用程式的版本更新而作異動，所以系統的維護成本高。

5.3.3 Microsoft BizTalk Server 2000 架構

本文提出的 XML 資訊服務架構這個理念，是現今資訊委外及企業強調協同合作所必然的趨勢。身為世界最大的軟體公司的微軟(Microsoft)，已經見到這個趨勢，因而在 2001 年初正式推出 BizTalk Server 2000。這個套裝軟體是用於交換商業資料的資料解譯和應用程式整合的伺服器程式，它使用 XML 技術以讓使用者可以跨產業、平台、作業系統或底層技術的方式，使企業可以與其夥伴建立合作關係。伺服器提供標準的閘道透過網際網路傳送和接收文件，並包含一些服務以確保資料的真確性、安全性的傳遞。

雖然由 Microsoft 主導的 BizTalk 組織，定義 BizTalk Framework 2.0，刊載一個利用 XML 描述的 BizTalk 商業文件標準，其內容為 SOAP 的擴充。但是 BizTalk Sever 2000 的主要用途不僅在處理 BizTalk 文件，它可以支援多種產業 XML 商務標準文件，因此 BizTalk Sever 2000 的核心不在 BizTalk Framework 本身。例如 BizTalk Sever 2000 可以配合 BizTalk Server Accelerator for RosettaNet 這個套裝元件，使這個伺服器具備處理 RosettaNet 標準文件的能力。RosettaNet 目前是資訊科技產業的主要商務標準，目前利用 BizTalk Sever 2000 實作 RosettaNet 標準最成功的案例，為我國的華碩電腦公司，它目前是全球最大的主機板製造商。華碩電腦公司利用 BizTalk Sever 2000 以及 Biztalk Server Accelerator for RosettaNet，成功達成供應鍊整合的目的。

Travis 在「XML and SOAP Programming for Biztalk Servers」一書中[3]，詳述 BizTalk 架構以及 BizTalk Sever 2000 在電子商務的應用。他提出的一個架構同樣是以 Web 資訊交換服務架構為基礎，在此茲將該架構稱作 BizTalk Server 2000 架構，以圖 5.12 加以表示。

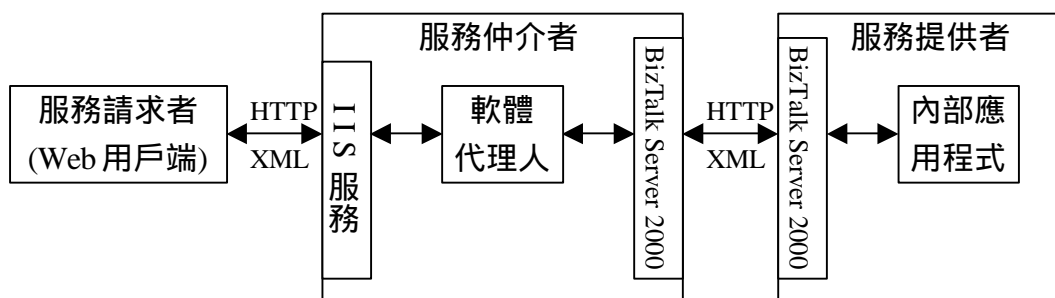


圖 5.12 Microsoft BizTalk Server 2000 架構

該架構的運作流程：服務仲介者中的 IIS(Internet Information Service) Web 服

務，接受到服務請求者的 HTTP 請求之後，隨即呼叫執行在內部的軟體代理人程式。軟體代理人根據其要求查詢提供該服務的服務提供者資訊，連同此 HTTP 請求，將之送至 BizTalk Server 2000。此伺服器將用戶的請求轉譯成 BizTalk 標準的文件，並將此文件傳送至服務提供者平台上的 BizTalk Server 2000。服務提供者平台上的 BizTalk Server 2000 將此 BizTalk 文件轉譯成內部應用程式可以接受格式的文件，並將之送至內部應用程式進行處理。處理完畢後將傳回 BizTalk Server 2000，再傳回服務仲介者平台完成回應處理。

架構評論：

微軟恃其在軟體研發及作業系統、Web 伺服器及瀏覽器獨佔的優勢，推行 BizTalk 標準以及行銷 BizTalk Server 2000，將會有不錯的市場接受度。而 BizTalk Server 2000 家族的支援產品有許多，可以克服許多在企業或系統整合上的問題。唯該伺服器承繼微軟其他的產品，大多僅能在微軟開發的作業系統上執行。在導入微軟伺服器作業系統及該伺服器成本昂貴的因素考量下，將造成許多欲導入電子化的小型企業的障礙。另外，上個架構所提的二個缺點：同步回應模式過於繁雜以及版本更新的問題，在該架構同樣無法克服。

5.4 小結

討論上述架構的目的，在為下一章將設計的类型錄搜尋代理人架構，進行方案的評估選擇，並作為架構改良的參考依據。將上述四種架構作一個整體歸納及評論，我們發現行動代理人架構，是一個極佳的分散式運算架構，它比起後述三種架構還更有效率。如果套用在第三章中所談論的分散式運算架構的分析可以發現，後述三種以三層式 XML 建構的系統架構，如果單純從服務仲介者及服務提

供者雙方來看，它是一個二層式架構的應用，而這三個架構在此使用的分散式運算架構是主從式架構，也可以稱為傳統的遠端程序呼叫(RPC)架構。我們將服務仲介者視為服務提供者的用戶端，而服務提供者為伺服器端。程式碼、執行環境及資源皆在服務提供者平台上，服務仲介者是透過一個用戶端程式透過 HTTP 請求的方式，遠端傳輸參數及呼叫執行程式。由於服務仲介者及提供者二者之間使用主從式架構，因此會有版本更新的問題。一旦伺服器程式需要更新，則用戶端程式必然需要重新改版建構。另外，第三種及第四種架構提供一個極佳的應用程式整合架構，但是應用在型錄搜尋代理人的架構中，反而過於複雜。因為身為服務提供者的供應商提供的型錄服務，利用 Web 來提供型錄服務。所以擷取供應商型錄最有快且有效的管道是擷取供應商的 Web 型錄服務，而非與供應商內部的應用程式作整合。因而應用在型錄搜尋代理人架構中，不建議使用這類複雜的架構。

因此提出一個改良架構的想法，其目的在解決上述四個架構中面臨的困難，分別為資料庫維護成本高且應用缺乏彈性、網路資源負荷過重、程式版本維護不易等問題。整合這類架構的優點，利用協同合作的架構機制，使得型錄搜尋代理人的構想得以實作。

第六章 型錄搜尋代理人架構設計

由於電子交易市集是一個發展尚未成熟的領域，所以並沒有一套具公信力的運作模式可以完全作為經營者參考。不論運作模式為何，電子交易市集必然要扮演資訊服務仲介者的角色，以協助買賣雙方進行供需配對或是完成線上交易。其中協助買賣雙方進行供需配對是電子市集的基礎必備功能，這包含替顧客搜尋供應商以及替供應商尋找顧客。這裡將對如何設計一個電子交易市集平台，以支援顧客搜尋供應商著手。要瞭解電子交易市集如何支援顧客尋找供應商，應從購買者的購買決策過程進行瞭解，並分析如何應用軟體代理人來協助顧客制定購買決定，這部分在第一節「購買決策過程分析」中進行討論。電子交易市集提供的「商品型錄代理」，是協助顧客制定購買決策的支援系統，這部分在第二節「型錄代理模式的電子交易市集」中進行討論。在前面三章我們針對行動代理人以及 XML 的資訊架構進行研究，並舉列以行動代理人設計的電子交易市集，以及電子交易市集相關的 XML 產業標準。值得一提的，雖然行動代理人提供一個穩固的分散式平台來支援電子交易市集，以實作企業間商務資訊的訊息交換及整合。但是建置成本高以及系統缺乏延展性，是利用行動代理人架構開發電子交易市集最大的阻礙。而 XML 是協助應用系統進行整合、訊息交換的語言，電子交易市集藉以實作存在不同企業中不同應用系統完成整合，以利供應商與顧客進行商務資訊交換。在 XML 標準尚未建立之前，許多學者設計的電子交易市集是以行動代理人技術為基礎，例如：使用 IBM Aglets 行動代理人系統開發的 TabiCan，它是一個提供航空訂票以及旅行住宿服務的電子交易市集[6]。以及在第五章中介紹，由 Sooho[27]等學者所提出一個應用行動代理人技術的電子交易市集架構。我們認為 XML 標準建立完成後，行動代理人架構不再是一個最佳的系統整合解決方案。但是在要求應用程式的服務品質、效率及非同步傳輸模式時，行動代理人系統最能夠提供一個有效的應用程式執行環境。使用 XML 架構為系統整合方案，並配合行動代理人系統建構，此資訊技術整合之架構在第三、四節作設計及探討。

6.1 購買決策過程分析

產品的顧客包含消費者及企業，但是消費者與企業的購買決策行為不致相同。消費者購買決策以個人偏好為重，而企業購買決策涉及企業自訂的採購流程，並藉由後端的採購及供應鏈管理軟體支援其流程，所以後者較前者複雜甚多。利用純粹 Web 平台的資訊技術，只能幫助顧客作資訊的蒐集，而分析過濾資訊的工作需要人為加以處理，這對顧客來說是既無效率，且人力成本不低的作業流程。

軟體代理人技術強調可以代理使用者的身份完成特定任務，並且完全不需要人為介入。因此，顧客的購買決策確實可運用軟體代理人技術來支援其決策過程。為將傳統的購買決策過程予以自動化，首先需要分析顧客制定購買決策的過程。依據 Maes, Guttman, Moukas 等人在“Agents That Buy and Sell” [22]一文中的定義，顧客購買決策過程共可以分成以下六個階段：

1. 需求確認階段：產品的資訊可以刺激顧客的需求，進而引起顧客的購買動機。
2. 產品篩選階段：包含資訊的蒐集及幫助顧客要買什麼產品，將顧客的對產品的需求具體化成產品的選擇。資訊的蒐集是建立在顧客提供的標準上進行產品特徵分析，以評估產品是否滿足顧客之所需，產生符合顧客預期標準的產品資訊。
3. 供應商選階段供依照上一階段的產品資訊，並且對供應商及產品本身進行評估。標準是建置立在顧客設定的條件參數(如價格、擔保、數量、送達時間、聲譽等)，產生符合顧客預期的供應商

資訊。

4. 議價協商階段：議價是顧客及供應商選定雙方同意的價格，以進行雙邊交易。拍賣會以及交易所是履行議價的主要模式。在一般的零售市場中，價格通常都是固定的。但是在其他市場中，比如股票、汽車、藝術品，價格以及其他交易事項的協商是整合在購買過程之中。對於價格或其他交易條件固定不變的產品市場，這個階段是可以省略的。經過議價協商的過程後，將產生特定可以符合交易條件的供應商。
5. 付款及運送階段：顧客與特定的供應商決定進行交易，因而進一步討論付款及運送的方式。上述各階段在完成交易協商的資訊流，而本階段引入金流及物流，完成交易處理。
6. 產品服務及評估階段：包含產品售後服務、顧客服務以及整個購買經驗及決策的滿意度評估。

這六個階段可以解釋消費者購買決策制定，其資訊蒐整的過程。應用於企業的購買決策雖過於簡化，但卻能表現其核心類似觀念。分析這購買決策的六個階段可以發現，使用軟體代理人技術可以代理顧客完成其中的產品篩選、供應商篩選、議價協商、付款及運送階段等四個階段[23]。因為需求確認階段在介紹產品基本資訊，利用現今的 Web 技術即能達到預期的效果。而產品服務及評估階段強調的是顧客消費的感受及經驗，這可以透過以該產品或供應商為主題的虛擬社群來滿足，但是效果有限不若顧客實際消費來得有效。現在對軟體代理人技術對這四個購買階段的貢獻進行分析討論。在產品篩選階段方面，軟體代理人依據顧客指定的產品特徵值，尋找符合其需求的產品。在供應商篩選階段，上個階段由軟體代理人所作的產品資訊蒐集的結果，在本階段作評估選擇。軟體代理人為顧

客過濾選擇適合的供應商。在議價協商階段，軟體代理人在協商過程中扮演居中者。由於有些供應商提供的產品並沒有固定價格，而是透過動態議價程式來決定價格，產品最終的價格為顧客與供應商雙方同意的結果。透過軟體代理人所採集的產品價格資訊，為顧客及供應商建立一個自動化的議價流程。最後在付款及運送階段，產品的運送及貨到交款是相伴而來的交易履行過程。利用密碼學技術，比如電子簽章，可以確保雙方署過的交易訊息無法被否認。一個協助顧客及供應商加解密支援的軟體代理人，為購後行為賦予保障。

利用軟體代理人的技術可以促使原有顧客的購買決策過程進行自動化。購買決策自動化，強調其能促使購買成本降低及效益提升。購買成本降低是因為利用軟體代理人，不需要過多的人力來負責採購過程中各個階段的管理，以及軟體代理人代理顧客找到最經濟實惠的產品。效益提升是因為軟體代理人，提供顧客的購買決策支援提升購買決策品質，加上購買決策過程透過資訊技術而透明化，避免企業未經核可的採購行為。

6.2 商品型錄服務與購買決策支援

Ariba 及 CommerceOne 是開發銷售採購自動化軟體起家的公司，這二家公司發現，企業採購自動化是企業對企業電子商務中非常重要的市場，尤其是在非生產性物料方面。企業在非生產性物料的採購，相較於生產性物料是較無排程規劃，採購次數多以及單筆價格低。但是端賴企業內部的採購自動化系統，並無法提供豐富的產品供應資訊，以提供採購自動化軟體進行資訊採集。因此，這二家公司著手推動電子交易市集的成立，集合多個顧客加強買方力量，促使更多的供應商加入採購自動化所構成的虛擬企業社群，這個虛擬企業社群即為電子交易市集。

所以早期多數電子交易市集成立的目標，在提供商品型錄以作為採購自動化軟體的資料來源。商品型錄至少需要包含產品描述、數量、價格、供應商等資訊，所以電子交易市集提供的商品型錄，至少支援購買者購買決策過程的前三個階段。這類市集將購買決策過程的後續三個階段，交由供應商與顧客雙邊處理。

一個提供商品型錄的電子交易市集，為配合企業採購行為而成立。此運作方式是早期電子交易市集的營運模式，有多數的人認為此模式是最有效的市場機制。雖然此模式猶如供應商／商品的入口網站，本身是資訊內容提供者，不提供交易機制。但是一個有架構且分類明確的商品型錄，是大部分供應商及顧客極需要的，特別是在非生產性物料的採購上。反而提供交易機制的電子交易市集，如提供議價功能，致使供應商削價競爭；亦或是提供金融服務而收取交易仲介費用，反而是多數企業所不期待的運作架構。

電子交易市集提供顧客線上的商品型錄，供顧客進行搜尋參考，以撮合供應商及顧客進行交易。這個模式最困難的地方是電子交易市集如何取得供應商提供的線上即時商品型錄。在此提出二個解決方案以討論電子交易市集如何整合供應商的資訊平台，提供顧客所需的線上即時商品型錄內容。第一個解決方案，筆者將之稱作「集中式的商品型錄管理模式」，參考圖 6.1 所示。第二個解決方案，筆者將之稱作「分散式的商品型錄管理模式」，參考圖 6.2 所示。其中，第一種模式「集中式商品型錄管理模式」，將商品型錄資料庫存放在電子交易市集平台中，並要求供應商定期或隨時上線維護各別的供應商型錄資訊。如此將需要供應商額外派駐人力維護其在電子交易市集中的型錄資訊，費時、成本高且易有人為疏失產生。即使供應商是用批次檔案上傳方式更新此資料庫，亦造成相當的網路頻寬負荷，更別說是在網路上傳輸造成資訊安全的潛在危機。相反的，第二種模式「分散式商品型錄管理模式」，擷取供應商原有資訊平台上所公佈的商品型錄內容，加以整理轉換進行再利用，而產生顧客所需要的資訊。所以不但不需要供

應商額外派任人員維護線上資料庫，亦可以避免人為輸入的疏失，以及批次資料傳輸造成大量資料揭露的安全問題。

分散式商品型錄管理模式，從軟體代理人的學術領域來看，它本身是一個提供顧客及供應商型錄代管的機制，這個機制筆者將之稱作「型錄搜尋代理人架構」，而分散式商品型錄管理模式是此架構的雛型。在下一節中，將利用此模式為基礎設計一個以資訊系統整合為目的的電子交易市集型錄搜尋代理人架構。

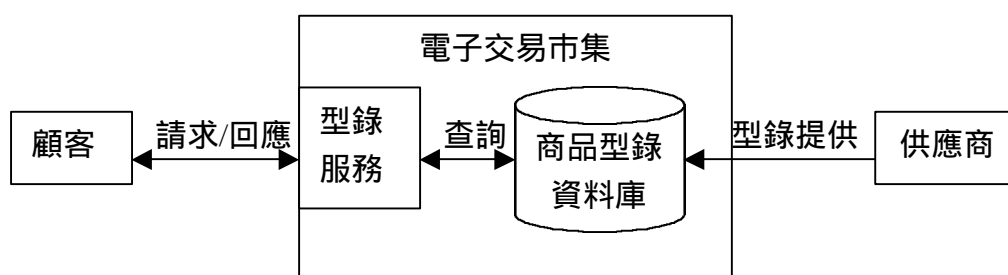


圖 6.1 集中式商品型錄管理模式

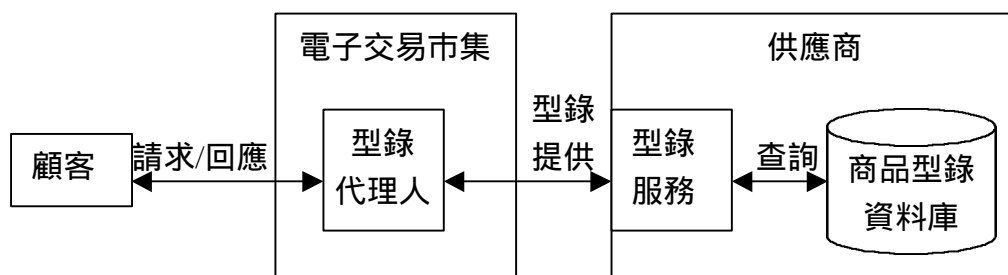


圖 6.2 分散式商品型錄管理模式(型錄搜尋代理人架構雛型)

利用行動代理人架構，我們可以設計一個以型錄搜尋代理人模式運作的電子交易市集，例如 5.2 節所述以行動代理人為基礎的電子交易市集架構。但是使用行動代理人架構有三個先天的限制，是應用它作為電子交易市集架構最大的阻礙。分別是應用延展性，後端資料庫存取以及應用程式整合，分述如下：

1. 應用延展性：行動代理人架構允許隲著在遠端平台上，行動代理人只具備存取用戶定義的資料庫介面。所以應用在電子交易市集之中，必須強制供應商的商品相關資料以市集所定義的資料庫格式存放，亦或是安裝相同的資料庫管理軟體，才能讓行動代理人有存取遠端平台後端資料庫的能力。由於行動代理人可以存取的資源有限，加上參與者必須建置相同的資料庫邏輯結構供行動代理人存取，所以應用延展性欠佳。參考圖 6.3 所示。
2. 應用程式整合：多數企業皆有各自建置的供應鍊管理應用程式，來維護企業內的商品型錄資訊。為彌補行動代理人架構應用延展性欠佳的限制，如能建立行動代理人與供應商供應鍊管理應用程式，彼此訊息交換的機制，將能使供應商不需要另外建置維護一個重覆的商品型錄資料庫。此舉為一個理想的作法，唯行動代理人不具備與其他非行動代理人的應用程式溝通的能力，所以大部分情況下此法並不可行。參考圖 6.4 所示。
3. 後端資料庫連結：如果行動代理人無法與供應商的應用程式作訊息交換，又期望供應商不需要額外維護市集所定義的資料庫，那麼直接連結供應商後端的商品型錄資料庫將是唯一可行的方案。為擷取存放在遠端平台的供應商商品型錄資料庫，行動代理人必須具備連結供應商後端資料庫的知識 (know-how)，亦即資料庫連結介面。但是一旦市集中有眾多供應商，勢必要求電子交易市集可以派遣具備不同的資料庫連結介面的行動代理人，至各供應商平台存取不同類型的後端資料庫。這將是需要極高維護成本的作法。參考圖 6.5 所示。

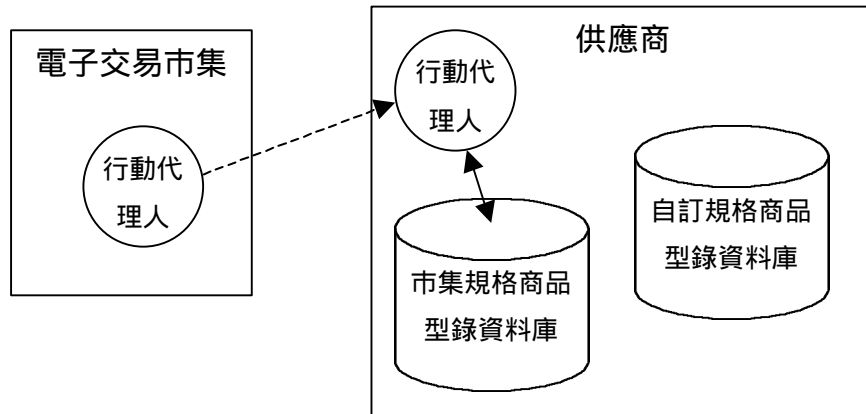


圖 6.3 市集定義資料庫連結

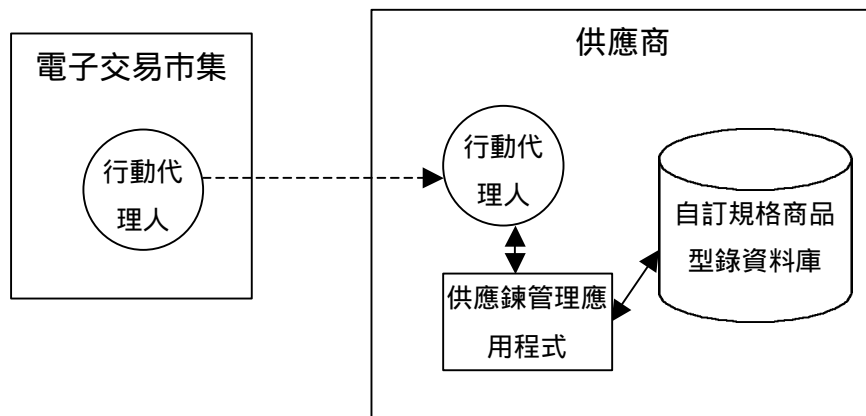


圖 6.4 應用程式整合

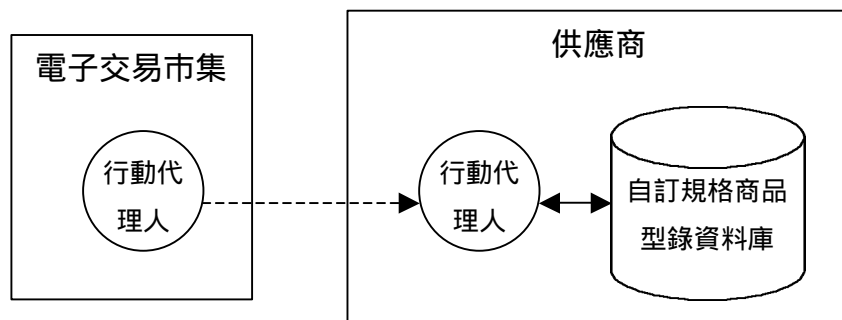


圖 6.5 供應商後端資料庫連結

為解決上述行動代理人應用在電子交易市集所產生的問題，這裡引進 XML 架構來改善原先行動代理人為基礎的電子交易市集，其運作上的缺失。以 XML 為基礎的電子交易市集，可以保留供應商既有的商品型錄資訊平台之架構。如此，可以大幅減低企業為參與電子交易市集，而需要付出的系統轉置及日後資料維護的額外成本。在以 Web 為基礎的 XML 架構下，扮演用戶端的顧客僅需安裝 Web 瀏覽器，可以作為使用者介面並與電子交易市集的應用程式作溝通。這個市集端的應用程式即為以 Web 為基礎的「型錄搜尋代理人架構」，它能夠接受顧客輸入的搜尋條件參數，然後觸發以事件為基礎的系統內部元件。透過內部元件的協同合作機制，將條件參數轉換成型錄搜尋代理人之具體任務，以完成商品型錄提供之目的。此具體任務為採集提供顧客所需商品之供應商的型錄。供應商公告其商品型錄的介面是 Web 伺服器，型錄資料內容以 XML 描述。此架構為參考先前提出的「分散式商品型錄管理模式」(圖 6.2)的進一步實作，實作的方式是供應商的型錄服務，是一個 Web 伺服器端的應用程式，依據電子交易市集傳送的商品型錄服務的 HTTP 請求，對後端的商品型錄資料庫進行查詢，並將結果以 XML 描述的商品型錄訊息加以回應。以圖 6.6 表示之。

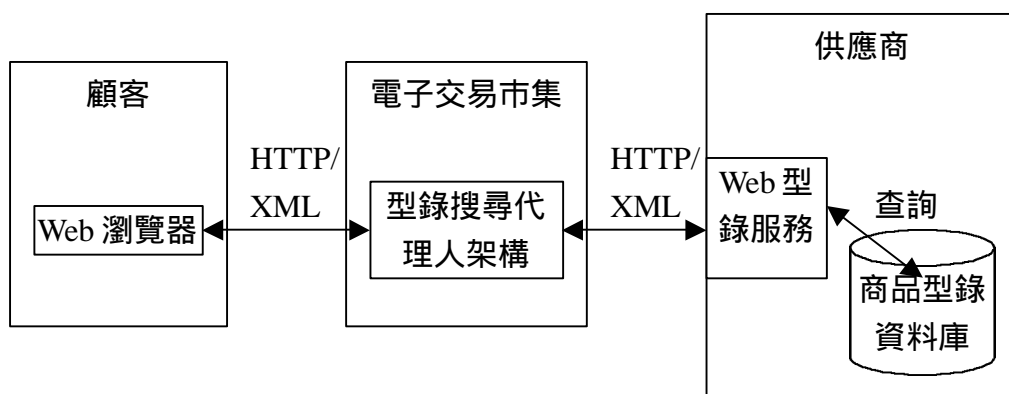


圖 6.6 以 Web 為基礎的型錄搜尋代理人架構模式

設計一個在電子交易市集的 Web 型錄搜尋代理人架構(參考圖 6.6)的目的，即在採集顧客所需之供應商商品型錄，各個供應商以 XML 來發布其商品型錄，這個商品型錄是使用不同的供應商自訂格式來表現，而型錄代理的目標即在將這些不同格式的商品型錄資訊，轉換成一個統一的格式標個文件中，作為回應顧客之資訊內容。這個系統應該具備四種功能，才能完成此目的。第一個功能為 Web 資訊服務的閘道，為將開發人員所設計的應用程式可以與 Web 伺服器進行整合，一個 Web 資訊服務的閘道是必然需要存在的，早期的方法是使用公共閘道介面(Common Gateway Interface, CGI)。第二個功能為行動代理人的機制，為求行動代理人技術可以實作，我們必須選用一套行動代理人系統，來達成系統引入行動代理人架構的目的。第三個功能為 XML 處理，在此稱作 XML 處理器。第四個功能為資料庫連結，即連結資料庫的機制。另外，一個市集自訂的格式標準，是回應顧客資訊的內容準則，在此選用「RosettaNet PIP 3A2」作為描述商品型錄的標準，使用它的目的在考量與其他市集建立溝通機制。首先，將介紹使用何種實作工具可以幫助系統具備此四種必備功能。然後，介紹「RosettaNet PIP 3A2」的標準內容。

本系統使用 Sun Microsystems 主導的 Java 作為程式開發語言，除了核心 Java 應用程式介面(API)之外，還需要配合 Sun Microsystems 以及 IBM 所提供的延伸應用程式介面，來實作本系統。這些延伸應用程式介面包含 Servlet、Aglets、XML for Java 及 JDBC，分別使系統具備 Web 資訊服務閘道、行動代理人、XML 處理器以及資料庫連結的功能。茲將此四部分詳述如下：

1. Servlet: Servlet 提供在 Web 伺服器開發應用程式上強而有力介面[18]，Servlet 是在 Java 虛擬機器上執行，由 Sun Microsystems 開發，具有安全及可攜性。許多 Web 伺服器開始具備 Servlet 支援，以提供 Web 開發人員 Servlet 執行環境。這裡的 Servlet 支援意指 Servlet 引擎，是 Servlet 的直譯裝置。我們可以

使用內建 Servlet 引擎的 Web 伺服器，如 Java Web Server，或是外掛 Servlet 引擎，使 Web 伺服器具備 Servlet 支援。本系統使用 Sun 的 Tomcat 系統，它可以用來當作內建 Servlet 引擎的 Web 伺服器，亦可以當作 Apache(Web 伺服器)的 Servlet 外掛引擎。利用 Servlet 開發 Web 伺服器的應用程式的優點，在於 Servlet 提供可攜性、效率、耐用性、安全性、使用方便、整合度高、延伸簡易，以及運用彈性等優勢。而本系統使用 Servlet 作為 Web 應用程式開發的工具，即在於它與 Java 應用程式高度的整合性。本系統採用 Web 瀏覽器作為用戶端介面，而 Servlet 本身可以提供 Web 瀏覽器與伺服器上的 Java 應用程式的訊息交換橋樑。具備 Servlet 引擎的 Web 伺服器可以接受用戶端的 HTTP 協定請求，然後呼叫 Servlet 程式內的執行緒，以執行處理此請求所需的 Java 應用程式。另外在系統安全方面，Servlet 函式庫可剖析在 SSL 連結上的 X.509 憑證。

2. Aglets：由 IBM 所提出的 Aglets 是一個行動代理人系統，從 Java 的角度來說，它同時也是一個 Java 的應用程式介面。我們可以利用 Aglets 應用程式介面所提供的函式庫，使我們可以利用物件繼承及介面實作的方法，來利用 Java 語言開發行動代理人應用程式。IBM Aglets 本身提供豐富的文件支援，使得開發行動代理人系統的學習障礙降低。當然完全相容於 Java 的特性，是本系統採用 Aglets 作為行動代理人系統開發的主因。
3. XML for Java：同樣由 IBM 開發的 XML for Java 是一個 XML 處理器，簡稱 XML4J。它可以讀取 XML 文件並且提供 XML 文件內容及結構存取的應用程式介面。XML4J 可以查核一個 XML 文件是否符合 XML 規範以及資料定義規則。本系統使用 XML 作為訊息交換的語言，並且利用 XML 文件作為資訊描述的邏輯結構。透過 XML4J 使得開發以 Java 為程式基礎的本系統，具備方便的 XML 文件及訊息交換功能。

4. Java Database Connectivity : Java 應用程式存取資料庫的介面叫做 Java Database Connectivity , 以下簡稱作 JDBC。關連式資料庫仍然是現今資料儲存的主要邏輯格式 , 資料庫管理系統提供方便的 SQL 敘述來對資料更新維護 , 它的功能是很難被其他技術取代的。往後多數的應用程式預期會使用 XML 文件來表示資料內容 , 但是資料儲存的邏輯格式仍然是關連式資料庫 , 即資料的搜尋及維護是透過 XML 文件為中介。JDBC 是 SQL 層次的 Java 應用程式介面 , 系統開發人員可以用它來執行 SQL 敘述 , 以查詢及維護資料庫內的資料。開發 JDBC 應用程式必須載入 JDBC 驅動程式 , 才能連結實際的資料庫。JDBC 驅動程式是由開發資料庫管理系統的供應商所提供 , 而目前市場上可見的資料庫管理系統 , 皆提供 JDBC 驅動程式。另外 , 在微軟的作業系統上 , 一般以定義 ODBC 的方式讓應用程式來存取資料庫 , 而利用 Sun 提供的 JDBC-ODBC 橋接驅動程式 , 可以讓應用程式透過它來存取 ODBC 定義的資料庫資源。

除了上述系統必備功能之外,建構一個符合未來相容性考量的電子交易市集文件結構,亦是相當重要的。在第四章中,我們簡述電子交易市集相關的 XML 商務標準有: OBI、eCO、RosettaNet、commerce XML 及 BizTalk 等。在此茲採用 RosettaNet 為本市集之 XML 商務標準,以期與外部企業作商務資訊交換。主要原因為後續一章,將探討此架構應用在個人電腦產業的電子交易市集上。而 RosettaNet 主要參與成員為世界級的資訊電腦大廠,比如 IBM, HP, Intel, Microsoft, 而其初期制定的標準在資訊科技、電子零件、半導體等三大產業上。

RosettaNet 建立三個主要的標準[25]: 參考辭典(Dictionary)、RosettaNet 實作架構(RosettaNet Implementation Framework, RNIF)、交易夥伴介面流程(Partner Interface Processes, PIPs)。其中參考辭典在形容作業或產品規格的詳細內容,

RosettaNet 實作架構定義傳輸訊息的規範，交易夥伴介面定義一致化的商業流程，透過此流程以期使企業與其交易夥伴，可以藉此流程進行商務訊息交換，這個訊息是使用 XML 為描述語言。

RosettaNet 定義的企業夥伴介面流程共區分成七個主要的群集(cluster)：編號從 cluster 0 至 cluster 6，這七個群集分為有數個部分(segment)，每個部分皆有數個子集，每個子集我們稱作一個 PIP。在本系統中，由於目的為商品型錄代理，所以我們需要的是電子交易市集，可以提供即時的供應商商品相關資訊(可提供價格及數量)。在 cluster 3(訂單管理)的 segment A(報價及訂單登記)中的第二個流程「商品價格及數量查詢」，即所謂的「PIP3A2」，是 RosettaNet 定義商品型錄的標準。以 PIP3A2 標準規範電子交易市集的型錄內容，其好處是未來應用程式可擴充性高，且易整合其他企業的應用程式，主因是未來資訊產業的商業流程訊息內容將朝向 PIP 的標準規範。

6.3 型錄搜尋代理人架構設計

先前在第四章曾引述，以瀏覽器及 Web 伺服器所構成的 Web 架構，可以運用在任何分散式運算架構之上。為架構「型錄搜尋代理人架構」，筆者將 Web 架構運用在三層式以及行動代理人二種分散式運算架構上。利用三層式架構，我們可以設計一個同步即時的 Web 應用程式，而行動代理人架構，則能協助我們在系統中引入非同步的運作模式。所以本系統區分成二種模式，第一個模式為「型錄搜尋代理人架構同步回應模式」，第二個模式為「型錄搜尋代理人架構非同步回應模式」。

顧客使用 Web 瀏覽器，透過 HTTP 協定向存在於電子交易市集中的「Servlet 介面程式」提出請求，這個型錄代理的過程，由觸發 Servlet 介面程式而啟動。這個 HTTP 協定的訊息內容為顧客所要搜尋的商品型錄條件，包含有：產品類別、價格、數量以及有效時限等基本參數。在顧客輸入的商品型錄的條件參數中，如果回應有效時限為「即時」，則啟動「型錄搜尋代理人架構同步回應模式」。如果有效時限為限定之時間(如一週內)，則除了啟動商品型錄代理同步回應模式之外，還需要啟動「型錄搜尋代理人架構非同步回應模式」。

在此茲將「型錄搜尋代理人架構」的實際細部運作架構以圖 6.7 表示，其運作過程如下所述：

商品型錄代理同步回應模式(三層式架構模式；靜態式代理人架構模式)

1. Servlet 介面程式，依據顧客輸入的產品類別，查詢提供此類產品的「供應商資料庫」，然後傳回這些供應商的商品型錄所在的 HTTP 協定網址(URL)，例如：<http://www.supplier.com>。
2. 依據供應商的商品型錄網址，作為「型錄蒐集程式」的輸入參數，型錄蒐集程式開始依據這些網址，去採集這些供應商的商品型錄資訊。這個程式可以主動使用 HTTP 協定，同時向多個供應商的 Web 伺服器提出請求，並非傳統使用瀏覽器啟動 HTTP 的請求與回應。
3. 供應商平台上的「Web 型錄服務程式」，接受電子交易市集所傳來的 HTTP 請求。依據請求的參數對後端的商品型錄資料庫，進行商品型錄的查詢，然後產生 XML 描述的商品型錄內容，回應至來源地 - 型錄蒐集程式。

4. 型錄蒐集程式接受到以 XML 描述的商品型錄內容後，隨即載入「型錄服務轉換規則」，將供應商提供的商品型錄格式，轉換成 RosettaNet PIP3A2 所定義的標準格式。然後判斷是否所有供應商皆已送回商品型錄資訊，如果還未全部送回則繼續處理本步驟，如果已經全部送達，則繼續下一個步驟。
5. 蒐集所有提供該產品類別的供應商商品型錄資訊之後，「型錄代理回應程式」載入「型錄代理文件規則」，這個規則描述上個步驟所轉換完成的商品型錄內容，放置在 PIP3A2 回應文件中的相對位置。本程式在此判斷各個供應商所提供的商品型錄是否符合顧客預期的價格或數量，然後將符合條件的商品型錄放置在型錄代理的回應文件之中，這個文件以 PIP3A2 規範的 XML 語法加以描述。
6. 回應文件中指定其 XSL(eXtensible Stylesheet Language)的位置，XSL 是針對 XML 文件所設計的排版樣本，本身亦由可自我描述的 XML 加以規範，目的在使用於描述資料內容的 XML 文件，可以依此樣本而展示在用戶端的瀏覽器之中，猶如 HTML 在瀏覽器的效果。所以本步驟將商品型錄回應文件與其排版樣本 XSL，一起透過 Servlet 介面程式傳回至顧客平台上的 Web 瀏覽器。
7. 顧客透過瀏覽器可以看到商品型錄資訊，包含產品名稱、價格、數量以及供應商資訊等。供應商資訊包含該項產品的交易網址，顧客可以直接點選該網址而與供應商進行交易。

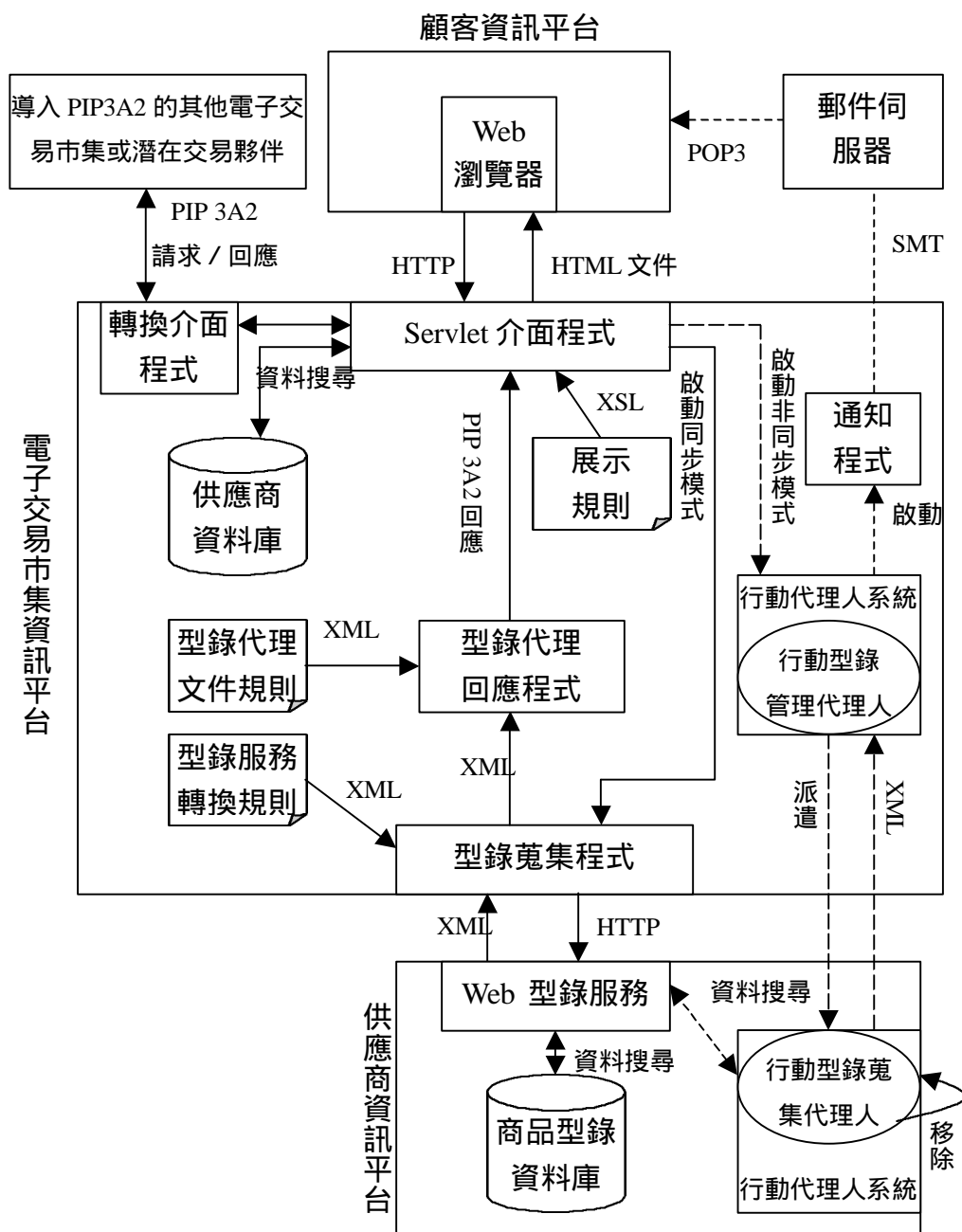
商品型錄代理非同步回應模式(行動代理人架構模式)

1. Servlet 介面程式，依據顧客輸入的產品類別，查詢提供此類產品的「供應商資料庫」，然後傳回這些供應商的商品型錄所在的行動代理人協定的位址，由於本系統使用 Aglets 作為行動代理人系統，其使用的代理人傳輸協定稱作 ATP(Agent Transport Protocol)，例如：atp://www.supplier.com。
2. Servlet 介面程式將顧客設定型錄條件連同供應商 ATP 位址，送至電子交易市集平台上的行動代理人系統中的「行動型錄管理代理人」。該行動代理人依據顧客設定型錄條件作為參數產生「行動型錄蒐集代理人」，並送至供應商 ATP 位址所在的行動代理人系統。
3. 與以往行動代理人建構的系統不同，在遠端供應商平台上執行的行動代理人採集的資料來源，不是固定格式的資料庫，而是 Web 型錄服務程式所公告的 XML 商品型錄資訊。在這裡行動型錄蒐集代理人會以固定時間為週期去更新查詢這個資訊，一旦日後供應商所提供的商品能夠符合顧客期望的條件，這個行動代理人就會主動自我移除，並且傳送 XML 描述的商品型錄，至電子交易市集的行動代理人系統中的行動型錄管理代理人。否則在有效時間內，行動型錄蒐集代理人會持續執行本步驟。
4. 電子交易市集上的行動代理人系統，接收到行動型錄蒐集代理人回應的商品型錄資訊後，執行一個「通知程式」，利用 SMTP 傳送電子郵件，至顧客電子郵件所在的郵件伺服器，待日後顧客利用電子郵件用戶端系統(例如 Microsoft Outlook)，透過 POP3 等協定下載記載型錄資訊的電子郵件。

符合 PIP3A2 請求的回應模式

此架構的回應遵循 PIP3A2 回應標準，同樣此架構亦能接受 PIP3A2 的請求

訊息，然後啟動上述二個同步及非同步回應模式。在資訊科技、電子元件以及半導體產業中，多數企業或電子交易市集已經開發或尚在導入 RosettaNet 標準，所以提出一個轉換介面構想，這個介面程式接受其他電子交易市集或潛在交易夥伴的 PIP3A2 標準請求，然後轉換成一個 Servlet 介面程式的 HTTP 請求，等待此 Servlet 的回應，使得在不更改原有架構下，讓本架構同時也具備 RosettaNet 實作架構的功能。



*上圖的虛線代表非同步模式，實線代表同步模式

圖 6.7 電子交易市集之型錄搜尋代理人架構同步 / 非同步回應模式

利用 XML for Java(XML4J)提供的應用程式介面(API), 程式開發人員可以解決處理 XML 相關的問題, 包含: 剖析 XML 文件、產生 DOM tree、轉換 XML 文件等。在 Web 服務架構下, 我們配合 XML4J 處理此架構的同步回應模式, 不會造成執行上的障礙, 因為程式執行的環境皆在電子交易市集平台上, 而此平台上安裝 XML4J 的環境。但是處理行動代理人架構, 我們想要在供應商執行型錄蒐集的處理, 就必須考量供應商平台上是否具備型錄蒐集的處理能力。

從物件導向的觀點來說, 型錄蒐集程式透過物件繼承的方法承繼 XML4J 的類別特性, 亦即 XML4J 提供 Java 環境執行型錄蒐集程式必備的 API, 在 Java 的環境中稱作此 API 為類別檔。在供應商平台上, 行動蒐集代理人欲完成型錄蒐集的任務, 首先必須在該平台的 Java 環境中具備 XML4J 相關的 API。經由 Java 的檔案壓縮機制, 可以處理型錄蒐集所需要相關的 XML for Java 的類別檔壓縮成一個檔案, 如此一來可以解決整個 XML4J 的 API 檔案過大的問題。行動型錄蒐集代理人透過類別檔隨選架構以配合 API 處理, 如果供應商平台上已經安裝 XML4J 環境, 則蒐集程式僅需載入該平台上的相關 API 類別類, 以繼承執行 XML4J 的類別和功能, 否則將下載市集平台上所提供的 XML4J 類別檔, 參考圖 6.8 所示。

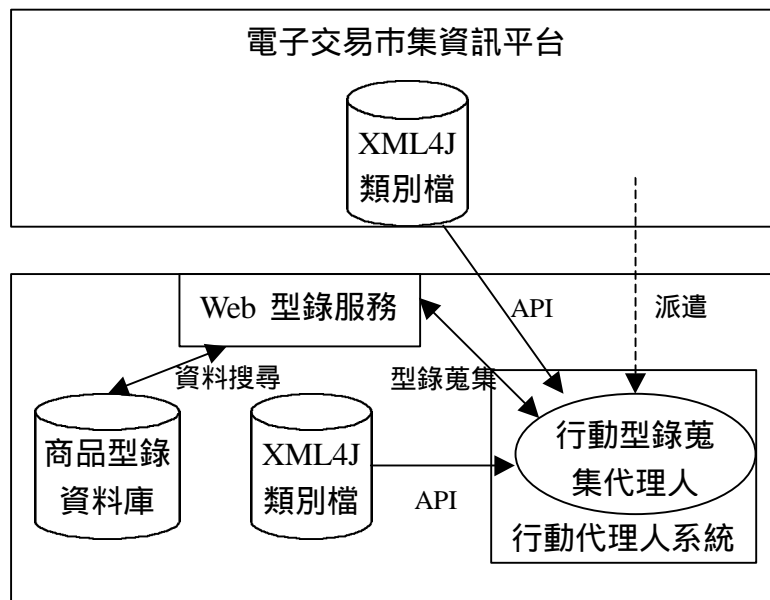


圖 6.8 行動型錄蒐集代理人類別檔隨選架構

6.4 在三層式 Web 平台上整合 XML/行動代理人的架構評價

上述筆者設計的型錄搜尋代理人架構,是一個在三層式 Web 架構上,以 XML 作為訊息傳輸文件規範,並導入行動代理人分散式運算架構的整合平台。現在將本架構與第五章所述的四個架構方案:行動代理人架構、XML/HTTP/Servlet 架構、XML/SOAP/Servlet 架構、BizTalk Server 2000 架構,作一詳實的比較,以瞭解本架構在電子交易市集中的型錄搜尋代理人,應用上的優勢及利基。茲將這個架構與上述四個架構應用在型錄搜尋代理人應用時,評量這個架構的優勢何在:

1. 與行動代理人架構比較

首先,一個純粹使用行動代理人架構的電子交易市集,將如同 5.2 節所述的架構,該架構有限制供應商應用彈性,以及需要額外維護型錄資料庫二大缺點,因此本架構僅在非同步回應模式中,使用行動代理人架構執行型錄蒐集的任務,阻斷純應用行動代理人架構的缺點。在本章第二節中亦對行動代理人架構的缺點作詳盡評估,因此在此不作重覆描述。

2. 與 XML/HTTP/Servlet 架構比較

本架構採用 XML/HTTP/Servlet 架構,來達成其中的同步回應模式。應用 XML/HTTP/Servlet 架構,可以讓我們跨平台運用 XML 訊息交換機制,達成系統整合以符合型錄搜尋代理人之需求。然而如果我們以 XML/HTTP/Servlet 架構,來設計一個型錄搜尋代理人架構的非同步回應模式,則運作模式將如圖 6.9 所示。很清楚地我們可以觀察到,由於電子交易市集上的型錄蒐集程式,需要定時

向遠端的供應商下載商品型錄資訊，將造成網路資料傳輸及系統效能上非常大的負載。例如設定三天內每隔一小時作一次商品型錄蒐集，那麼將需要作 72 次的資訊傳輸。然而使用行動代理人架構，型錄蒐集程式以行動代理人型式存在，並且在供應商的平台上執行，同樣是三天內每隔一小時作型錄蒐集，整個訊息的往來卻只有一次，所以相對來說其運作比三層式架構有更大的效益，參考圖 6.10 所示。

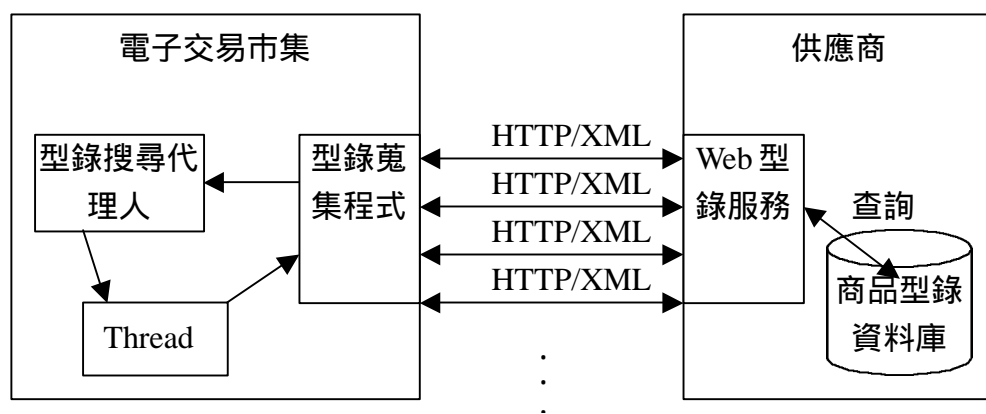


圖 6.9 XML/HTTP/Servlet 架構的型錄搜尋代理人非同步回應模式

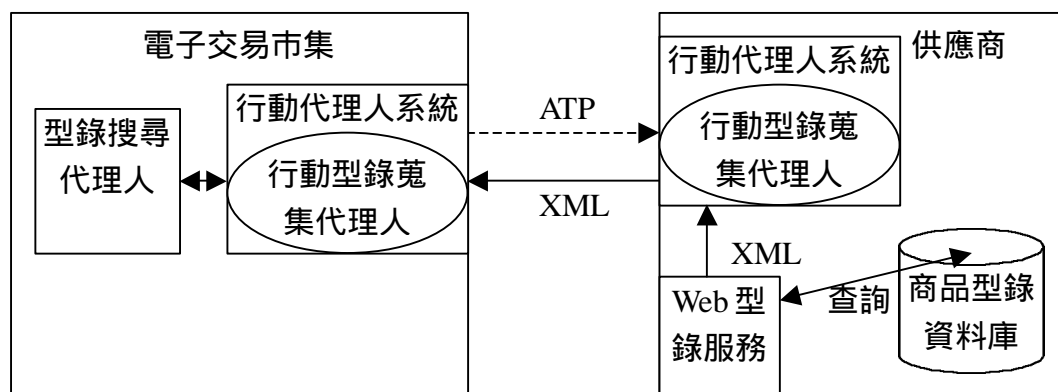


圖 6.10 行動代理人架構的型錄搜尋代理人非同步回應模式

3. 與 XML/SOAP/Servlet 架構比較

應用 XML/SOAP/Servlet 設計一個型錄搜尋代理人架構，其運作如圖 6.11 所

示。型錄搜尋代理人透過在同一個平台上的 SOAP 用戶端，向供應商平台上的 Servlet 利用 XML 文件提出 HTTP 請求，然後 Servlet 再將此 XML 轉送至 SOAP 服務。這裡的 SOAP 用戶端認定為型錄蒐集用戶端，而 SOAP 服務為型錄蒐集伺服器。SOAP 服務將此 XML 文件轉換成內部的型錄服務應用程式，可以接收輸入的文件。在同步回應模式中，SOAP 服務會接受到型錄服務應用程式的回應文件，然後再將文件送回電子交易市集平台。在非同步回應模式中，SOAP 服務將確認收到並且已經執行的訊息送回電子交易市集平台，等待型錄服務應用程式作一個時段的型錄資料庫查詢。這個架構解決了電子交易市集的型錄蒐集程式，與供應商的型錄服務程式整合的問題。即便如此這個架構與筆者設計的架構相較，有二個缺點。第一個缺點是 XML/SOAP/Servlet 架構處理同步和非同步回應模式的方法皆相同，但可以清楚發現處理同步回應模式 XML/SOAP/Servlet 架構顯然較本架構更為複雜且較無效率。第二個缺點是版本更新的問題，供應商平台上的 SOAP 服務與電子交易市集平台上的 SOAP 用戶端，必須是完全對應相容的。如果現在供應商平台上的型錄服務應用程式，其版本更變時則 SOAP 服務及用戶端皆必須一起變動。如此一來將不易掌握版本控制的問題，同樣維護成本亦不小。

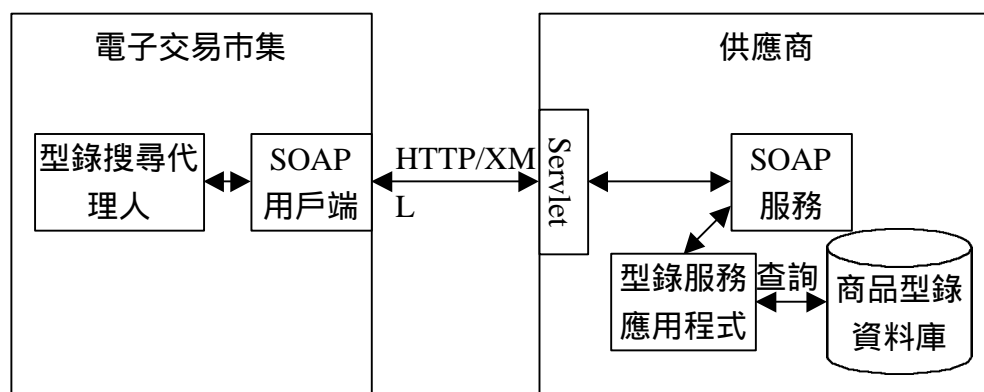


圖 6.11 XML/SOAP/Servlet 型錄搜尋代理人架構

4.與 Microsoft BizTalk Server 2000 架構比較

應用 BizTalk Server 2000 設計一個型錄搜尋代理人架構，其運作如圖 6.12 所示。由微軟研發的 BizTalk Server 2000 產品，促使二個不同平台的不同應用程式，可以透過 BizTalk Server 2000 文件轉譯及傳送的功能，達成系統整合機制。首先，電子交易市集平台上的型錄搜尋代理人，向在同一個平台上的 BizTalk Server 2000 提出一個請求，然後 BizTalk Server 2000 將此請求轉換成 XML 文件，再送至供應商平台上的 BizTalk Server 2000。然後此 BizTalk Server 2000 將此文件轉換成內部型錄服務應用程式可以接受輸入的文件。在同步回應模式中，BizTalk Server 2000 會接受到該應用程式的回應文件，然後再將此文件送回電子交易市集平台。在非同步回應模式中，BizTalk Server 2000 會送回一個確認收到且已經執行的訊息回電子交易市集平台，等待型錄服務應用程式作一個時段的型錄資料庫查詢。唯其架構僅能適用在微軟的伺服器作業系統上，無法在其他像是 Linux 的免費作業系統上運作。由於此限制，使得小型企業導入系統整合作業有較大的困難。再者，與 SOAP 的版本問題相同，電子交易市集平台以及供應商平台上的 BizTalk Server 2000 的應用程式必須相互配合，才能協同合作達成系統整合目的。而本論文提出的架構，則由於電子交易市集平台開啟行動代理人在供應商平台上執行，所以不需要供應商平台上的程式相互配合。

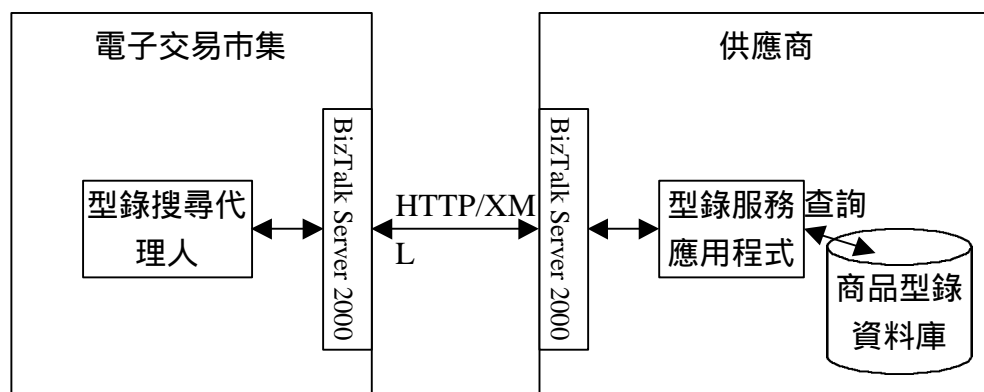


圖 6.12 Microsoft BizTalk Server 2000 型錄搜尋代理人架構

從上述的架構相互比較之後，可以發現本架構由於將三層式架構及行動代理人架構同時應用在 Web 平台上，可以彌補彼此在訊息傳輸與回應效能上之不足，而應用 Web 平台最大的優勢在於它的普及性最高，所以易於擴充系統作用範圍。三層式 Web 架構以 HTTP 協定為基礎，能夠以非連結同步模式傳輸資訊。此時用戶端在顧客平台中，中間層在電子交易市集平台中，而後端資料庫則存在供應商平台中。在三層式 Web 架構上，用戶端只需要安裝瀏覽器即能擷取所需資訊。其餘的工作，則交由電子交易市集平台與供應商平台上的應用程式協同合作。

傳統以行動代理人架構為基礎的電子交易市集[6][27]，要求電子交易市集，以及各個參與個體(包含企業或消費者)，皆必須安裝行動代理人系統。由於消費者對購物的便利性較重視，所以安裝行動代理人系統，將形成消費者參與的進入障礙。相反的，本架構只要求供應商及電子交易市集安裝行動代理人系統，提出商品型錄代理請求的顧客，僅需要使用現有的瀏覽器即可運作本架構之用戶端，這讓買方特別是消費者有更高的使用意願。

在一個產品的供應鏈體系中，通常至少包含四個參與主要個體，分別為供應商、製造商、零售商、消費者。供應商供給製造商生產產品所需的原物料，製造商將原物料生產組裝完成產品之後，將製成產品交託給零售商，最後零售商將產品銷售至消費者手中，整個過程稱為市場通路。電子商務的到來使得傳統零售商，必須轉型經營否則受到將相當的衝擊。本架構設計的電子交易市集是將供應商與製造商、製造商與零售商以及零售商與消費者，這四者之間的商務往來透過此市集提供的型錄搜尋代理人架構，來提供各個參與者購買決策支援服務。這個

架構的四個參與個體，與電子交易市集的關係以圖 6.13 加以表示。

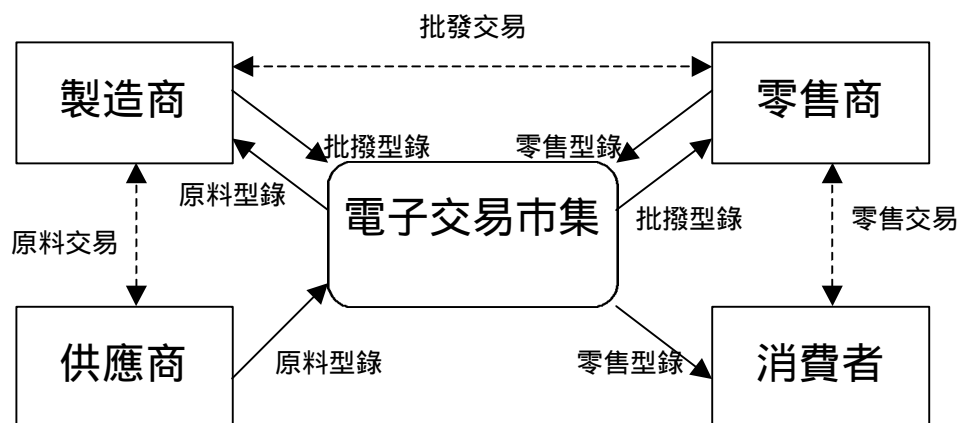


圖 6.13 電子交易市集與其參與個體

由於多數類別的商品，其賣方可以提供的商品數量以及商品價格是變動的，而非固定不變，特別是企業與企業的商品交易更是如此。商品的單位價格往往與交易條件相關，這些交易條件包含訂購數量、付款方式、到期日以及單方面事先協議的折扣等。所以商品型錄必須提供最新即時的資訊，才能正確提供購買者購買決策的參考。一個理想的電子交易市集架構，需要提供供應鍊所有參與者相關的商品型錄資訊，使得該產品的潛在買方及賣方皆能透過這個共通的型錄資訊，而即時制定購買決策，提升供應鍊之效率。

6.5 小結

目前主要的電子交易市集型態，多是由於某產品或產業屬於買方導向，買方

的市場趨力比賣方還要大，而組成的封閉式採購市集。然而供給該產品的供應商很多，而多廠競爭下廠商的品牌很難建立，所以賣方需要借由電子交易市集來擴增其市場的通路。這類市集既然是由買方趨力而形成，所以市集平台的設計，就必然以提供買方採購的便利性為首要，因而忽略賣方導入市集的彈性。在開放式的電子交易市集中，兼顧買方及賣方導入系統的利益及成本，為其成功的關鍵因素，這有賴行動代理人與 XML 技術配套完成。行動代理人系統提供電子交易市集平台自動化代理的功能，所以在電子交易市集的領域中是不可忽視的資訊技術。但是一個全然由行動代理人建構的電子交易市集，要求市集參與者必須妥善安裝市集定義的系統環境，有時還需要在原有系統作變更以遵循市集的規範，才能發揮功效。所以相當的不具有彈性，造成參與者進出的障礙。

以 Sooho[27]所設計的行動代理人電子交易市集架構為例，參與該電子交易市集的成員，必須在本地端安置符合其市集定義格式的型錄資料庫，方能使行動代理人擷取型錄資訊。雖說，利用資料庫管理領域中的技術，我們可以另外建置一個資料表，以關連對應原有的型錄資料庫中的資料表，因而不需要人力再行維護市集定義格式的型錄資料庫。但是，該架構的運作必須要求各地用戶平台安裝行動代理人系統，因而提高企業導入電子交易市集用戶平台的障礙。另外，將同步和非同步回應模式皆運作在行動代理人架構上，亦不是一個有效益的運作模式。因為行動代理人架構然提供一個非同步回應模式環境。但是利用它處理同步回應模式，反而較目前一般的三層式架構無效率。

因此本文提出行動代理人與 XML 這二個技術結合的架構，使得參與者可以彈性參加該市集，而沒有太多的限制。日後公眾式電子交易市集必然日益增多，市集平台的使用彈性將是成功與否的關鍵因素。

第七章 資訊產業應用與系統實作

前章是本論文針對電子交易市集應用所提出的架構概念。為求概念可以付諸實證，本章將先行探討在資訊產業上，電子交易市集的發展現況。然後，以個人電腦的電子採購為實例，說明如何利用此架構來實作一個資訊系統，以利個人電腦在電子交易市集進行採購型錄查詢。本資訊系統稱作電子交易市集「個人電腦型錄搜尋代理人系統」，以下簡稱作本系統。

7.1 系統的研究與探討

7.1.1 系統意義與目的

本系統在提供一個應用在資訊產業內，非交易性質的電子交易市集服務。資訊產品由於規格革新速度快加上價格變動迅速，使得資訊產品與其他電子類產品相同，具有產品生命週期短的特性。因此有效掌握客戶需求以及產品庫存對這個產業內的參與者來說，是相當重要的。資訊產業的電子交易市集，將可以讓參與者掌握客戶需求並能透過它增加客源，而供給廠商在供應鍊管理上，由於採接單後生產或接單後組裝，所以可以達到更佳的零組件或產品庫存管理。由於資訊產品項目極多，所以本系統茲以個人電腦(PC)的採購為例，來解釋說明系統的功能及實作流程。本系統建構的目的為：

1. 提供一個 PC 購買者及供應者彈性的市集參與平台：在 PC 產業的供應鍊體系上，不論是零組件或是 PC 系統的買方或賣方，皆為此市集的潛在參與者。

參與者可以在不更改原先的系統架構的基礎上，容易的進入或退出該市集。賣方企業不會因為加入此市集，而在原先的銷售系統上有所更變。主要做法是買賣雙方利用語言交換機制來作溝通，買賣雙方保留對產品的資料定義，轉換工作交由市集管理平台全權處理。設計系統的原則為最小化的用戶端設定及安裝。

2. 以 Web 架構之用戶端最小安裝為原則:一般使用行動代理人系統為架構基礎的電子交易市集，需要用戶端安裝行動代理人系統。這對於最終消費者用戶來說，是相當不方便的地方。所以本系統只需要進行採購 PC 的用戶安裝 Web 瀏覽器，這是由於 Web 瀏覽器的普及所致。
3. PC 型錄代理：PC 供應者所提供的 PC 數量、價格以及交易網址是 PC 購買者最關切的資訊。而本系統代理 PC 購買者，作 PC 型錄資訊的採集，以利其作採購決策。這裡所謂代理 PC 購買者採集型錄資訊，是指市集平台上的本系統，依據顧客設定的條件(產品類別、規格型號、預算單價、需求數量)為參數，進行 PC 型錄之採集及過濾，然後回應 PC 型錄資訊至購買者用戶端。購買者可以根據回應之型錄資訊決定交易對象，並點選供應商以進行交易。

7.1.2 我國資訊產業電子交易市集之市場機會評估

目前的資訊產品市場處於價格競爭的型態，以個人電腦(PC)為例，PC 的供給廠商主要分為二種。一種是提供組裝電腦(clone)的地區型電腦公司，這些電腦公司組裝的電腦是就地取材，品牌不具知名度，以低價位方式吸引顧客。另一

種是提供品牌電腦(brand)的 PC 大廠，例如 Compaq、Dell 及 IBM 等，以品牌知名度與售後服務吸引顧客，價格比組裝電腦居於劣勢。但在九十年末品牌電腦採低價模式，使得全球 PC 市場逐漸從組裝電腦佔多數，而變成現今以品牌電腦居多數。品牌電腦市場稱強的居面，造成許多以自有品牌零件提供組裝電腦所需的國內廠商，只能接取以美國為主的國外 PC 大廠的代工訂單才能生存。由於品牌電腦之所以能夠維持低價位，主因於其委託專業電腦製造商從事生產，以降低生產成本。如果國內電腦廠商持續採取接取國外代工訂單來經營市場，只會造成品牌電腦稱強的情況日漸嚴重，造成國內以自有品牌經營的電腦相關產品無法在市場上立足。

幸運地，組裝電腦之所以無法與品牌電腦競爭主因於其失去價格優勢。因為品牌電腦廠商委外代工所需的電腦零件成本，比組裝電腦需要從零售市場獲得的零件成本便宜。亦即品牌電腦是買方有議價主導權，所以賣方居於劣勢。而組裝電腦是賣方有議價主導權，所以買方居於劣勢。但是，一旦一個具全球性規模，屬於 PC 產業的公眾式電子交易市集出現時，預期將會提供所有 PC 廠商更公平的競爭機會。因為此電子交易市場是由多個買方與多個賣方組成，所以買賣雙方皆無法壟斷價格。這對於居於全球 PC 零組件(如主機板、晶片組、顯示器、機殼等)領導國的台灣來說，參與電子交易市集是相當重要的市場發展機會。

上述所言的內容，為筆者對於電子交易市集造就組裝 PC 的機會進行討論，特別是將來對台灣 PC 產業之貢獻。但能否有一個公正且公開的 PC 產業電子交易市集，可以具全球規模是值得注目的。不過在此之前品牌 PC 大廠所主導的國際大型電子交易市集，已經在台灣開始佈線，搶佔國內資訊電子相關的廠商作為其客戶。由品牌 PC 大廠 IBM 聯合 Ariba 和 i2 等廠商於 2000 年 6 月 7 日成立的 e2Open，吸收我國的品牌及主要代工 PC 大廠宏碁成為創始成員，並於 2001 年 1 月成立台灣分公司。另外，同為品牌 PC 大廠的康柏(Compaq)、惠普(HP)、捷威

(Gateway)以及我國的神通和大同等所共組的 Converge，原名為 eHitex。於去年併購著名的電子交易市集 Vertical 旗下，專營電子產業交易市集 NECX 之後，聲勢大漲並於 2001 年 4 月 11 在台灣成立分公司。這二個由產業龍頭所主導的電子交易市集，初期是屬於私有式電子交易市集型態。未來則將走向公眾式電子交易市集型態，以吸收更多的廠商加入其構成的社群之中。其中 Converge 在 2000 年的營收高達 1 億美金，與其資本額相當。營收來源有 60% 來自亞太地區，特別是台灣及大陸所構成的大中華地區。足見台灣恃其為世界資訊電子主要製造國之一的優勢，是其吸引從事此行業的電子交易市集廠商積極投入的主要因素。表 7.1 針對 e2Open 及 Converge 作一比較[32]。

	e2Open	Converge
創始成員	IBM、日立、松下、LG、北電網路、Seagate、旭電(Solectron)、東芝、朗訊與宏碁	安捷倫、超微、康柏、佳能、捷威(Gateway)、日立、惠普、SCI、旭電、Western Digital、神通、大同
資本額	2 億美金	1 億美金
採用解決方案	交易市集平台：Ariba 供應鍊管理：i2 產品設計管理：MatrixOne 零組件資料庫：PartMiner、Aspect 運籌管理：FlightMatrix、LogiMatrix	交易市集平台：Vertical 供應鍊管理：Manugistics 產品設計管理：SpinCircuit 零組件資料庫：PartMiner 運籌管理：Celarix 金流機制：escrow.com 整合平台：WebMethods

表 7.1 二大電子產業電子交易市集 e2Open 及 Converge 比較表[32]

問題是上述所言的 e2Open 以及 Converge，創始成員皆為資訊電子產品的主要採購商，其所經營的電子交易市集，可能有球員兼裁判之嫌。亦即電子交易市集的經營目的，可能為圖利己利而削弱供應商參與意願。然而一個由公正第三者成立的大型資訊電子產業的電子交易市集並未出現，即使出現雖然可以較易吸引供應商加入，但是品牌 PC 大廠已經成立其主導的市集所以未必會參與。

不論如何，一個公正的資訊電子產業的電子交易市集，對台灣的 PC 製造商及供應商來說，有絕對的助益。在國內，行政院於 1999 年 6 月 3 日通過經濟部的「產業自動化及電子化推動方案」中，選定資訊業為推動標竿，擬定「推動資訊業電子化計畫」，以期優先完成資訊業 B2B 電子商務示範體系。計畫時間自 1999 年 7 月 1 日至 2001 年 12 月 31 日止，分為 A 類及 B 類計畫。其中 A 計畫為促成資訊產品採購商，比如 IBM、康柏及惠普等，結合國內資訊業組成供應鍊體系三至五個。B 計畫則是促成國內資訊產品主導廠商結合國內重要零組件供應商，組成供應鍊體系 20 至 30 個，帶動 2500 家中小企業導入電子化作業能力。其中 A 計畫通過審核的廠商為 IBM、康柏、惠普共計三家。B 計畫通過審核的廠商則為神達、宏碁、大眾、華碩、台達電、英業達、微星、大同、新寶、致伸、倫飛、誠洲、華宇、華通、新寶等共計十五家。

基於資訊產業為政府推動產業自動化的先導產業，所以國內資訊業廠商將是供應鍊電子化最早成熟的企業，是發展國內資訊產業電子交易市集的利基。在發展電子交易市集之前，企業電子化是成功的先決因素。如果企業沒有完備的資訊基礎建設，以及產業運作的資訊交換標準，對於在產業內推動電子交易市集，特別是直接性生產物料的採購，將是推動上相當不利的阻礙。而由於資訊業電子化的 A、B 計畫，將於 2001 年底結案，所以未來發展資訊產業的電子交易市集將是一項利多。

另外，在第 2.6 節曾言，政府推行 TIM 以建立國際跨產業的國家級電子交易市集。此計畫中的配合措施第四項明訂：「配合產業自動化與電子化推動方案，鼓勵國內企業特別是中小企業能盡量應用網路電子市集的行銷管道。」另外，在推出方式的第三項則明訂：「產業主導成立重點產業別垂直式網路電子市集，政府於幕後協助推動。」足見政府在進行產業電子化，以建立企業供應鍊電子化、採購自動化的基礎建設之外，並鼓勵產業內成立電子交易市集，以期產業內的廠商可以透過市集建立有效的行銷管道。其中早期推動的資訊業電子化，已經促使資訊業廠商具有電子化作業能力。所以在資訊產業內推動電子交易市集，成為技術障礙將是最低的。無怪乎國際大型電子交易市集 e2Open 及 Converge，會積極搶佔台灣資訊產業電子交易市集的市場。

由於 e2Open 及 Converge 多為國際級的品牌資訊產品大廠，所以能夠吸引這些大廠的一階供應商加入。所以在 A 計畫中，由國際資訊產品採購商與國內的資訊產品領導廠商形成的供應鍊體系，其中的國內資訊產品領導廠商有參與 e2Open 及 Converge 的絕對動機。但是在 B 計畫中，由國內資訊產品領導廠商與其為數眾多的中小型供應商形成的供應鍊體系，其中的眾多中小型供應商沒有必然參與 e2Open 及 Converge 的動機。也因此 e2Open 及 Converge 在台成立之初，即未把國內眾多的中小型供應商納入其客戶。在這裡可以發現，想要在國內創新成立一個資訊產業電子交易市集，其機會是在 B 計畫中的供應鍊體系，而非投入 A 計畫已經由大型電子交易市集所經營的供應鍊體系。關於資訊產業的供應鍊體系，配合資訊產業電子化的 A、B 計畫，以及 e2Open、Converge 電子交易市集，來歸納出國內資訊產業電子交易市集之潛在機會，參考圖 7.1 所示。由於目前在 B 計畫領域內的供應鍊體系，沒有一個具規模的電子交易市集出現。預料一個由公正第三者經營，且參與者導入作業平台成本低的公眾式電子交易市集，將相較於 e2Open 及 Converge 有更大的潛在機會。因為 B 計畫供應鍊體系的廠商多為中小型企業，本身與國際品牌資訊產品採購商沒有直接利害關係，再加

上沒有大型企業有高預算建置電子化作業平台。因此相信本系統強調導入容易且建置成本低低的特性，必能適用在 B 計畫領域內的公眾式電子交易市集。

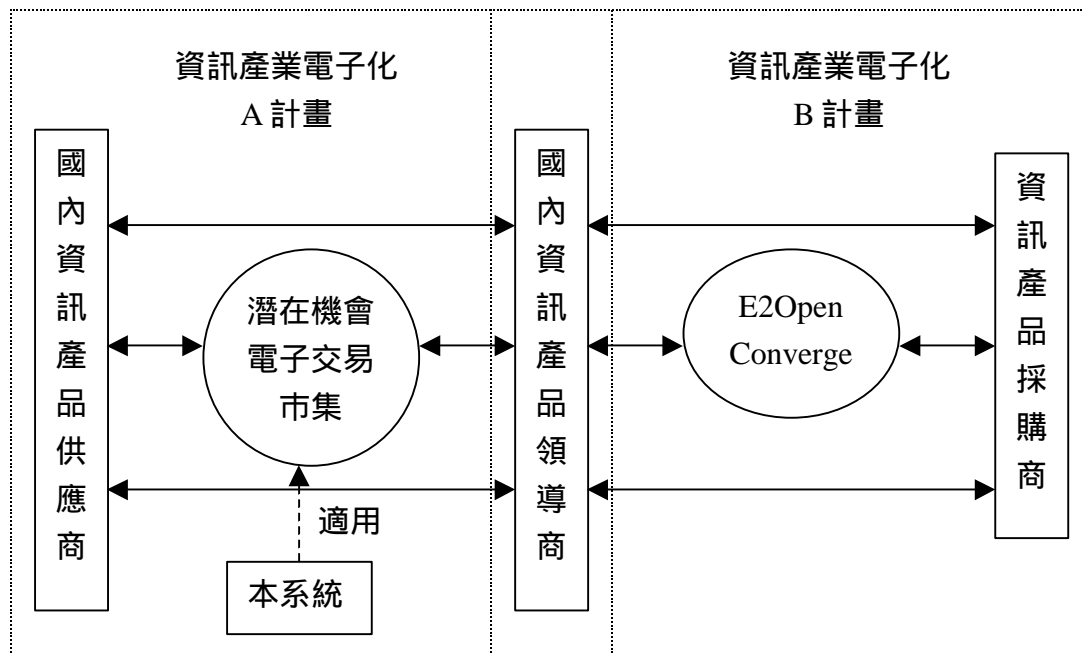


圖 7.1 本系統適用之市集

另外，經濟部推動資訊業電子化，其中 B 計畫是我國 15 家資訊產品主導廠商所共同組成，其使用的產業資訊交換標準，在列出 RosettaNet、EB\XML、Biztalk 等 XML 電子商務標準進行討論之後，最後決定以 RosettaNet 作為 B 計畫中，資訊產品主導廠商與其供應商的資訊業商務交換標準。但是平均導入 RosettaNet 的金額從五百萬至一千萬不等，對許多小型供應商造成極大負擔。所以本應用於個人電腦採購的電子交易市集系統，利用型錄文件格式交換技術，使得供應商可以不必具備 RosettaNet 流程處理能力。即使本系統只用到 RosettaNet 標準中的一小部分，但是可預期利用 RosettaNet 代理文件處理的構想，設計一個電子交易市集平台，將促使眾多的廠商有參與電子交易市集的意願。

到底資訊產業的電子交易市集有多大市場潛力，會不會經營資訊產業電子交易市集易於泡沫化，這是大眾最值得關切的議題。以樂觀的角度來說其實不會，

原因是 Forrester Research 在 2000 年 4 月評鑑及預估今後最有成長潛力發展電子交易市集的產業的調查中發現，此產業在 2004 年將有 4370 億美金的年交易額，是所有產業的第一名，參考表 7.2 所示[35]。

產業類別(單位:10 億美金)	2000 年	2004 年
電腦與電子零件(Computer and electronics)	\$13.9	\$437.0
能源(Utility)	\$29.7	\$243.0
汽車(Motor Vehicles)	\$7.0	\$210.6
石油化學(Petrochemicals)	\$1.6	\$147.1
紙張與辦公室用品(Paper and office products)	\$0.4	\$138.1
食品與農產品(Food and agriculture)	\$0.7	\$56.5
運輸與倉儲(Shipping and warehousing)	\$0.2	\$45.1
建築(Construction)	\$0.1	\$43.7
藥品與醫療用品(Pharmaceutical & medical products)	\$0.1	\$34.1

表 7.2 各大產業電子交易市集的成長潛力預測表[35]

7.3 系統實作

本系統的主要參與個體為電子交易市集以及市集參與者，市集參與者分為 PC 供應商、PC 零售商及 PC 消費者(包含企業用戶及個人消費者)。電子交易市集提供本系統伺服器平台，而參與者是伺服器平台的用戶，參與市集的目的在蒐集 PC 產品以及相關零組件之線上型錄，以供交易決策之參考。PC 供應商提供 PC 系統及相關零組件，包含 CPU、主機板、記憶體、硬碟、光碟機、顯示卡、音效卡、顯示器等。PC 零售商負責 PC 零組件或是代理組裝 PC 銷售。PC 消費者是 PC 零售商的顧客，包含企業採購辦公室設備的用戶，以及個別單件或少量購買的消費者。在這裡電子交易市集中，PC 供應商扮演 PC 零售商的賣方。PC

零售商是 PC 供應商的買方，同時是 PC 消費者的賣方。PC 消費者則是 PC 零售商的買方。電子交易市集需要建置型錄搜尋代理人伺服器平台，而買賣方則需要建置代理人用戶端平台，其架構基礎如第六章所述。

PC 或是 PC 的元件(零組件)，可以建立一個樹狀結構圖來表示其產品的內容及關連性，參考圖 7.2。市集的交易標的分成 PC 以及其相關元件，這裡稱為產品項目。其底下又細分三個子集，一為規格，是此產品的主要類別型態，例如監視器的規格為尺寸，而光碟機的規格為多少倍速。二為元件項目，代表組成此產品的構成元件，元件本身繼承產品類別的所有特性。三為價格，價格是商品的單位動態價格，其隨訂購數量等因素而不同。四為數量，是可供給該產品的數量。

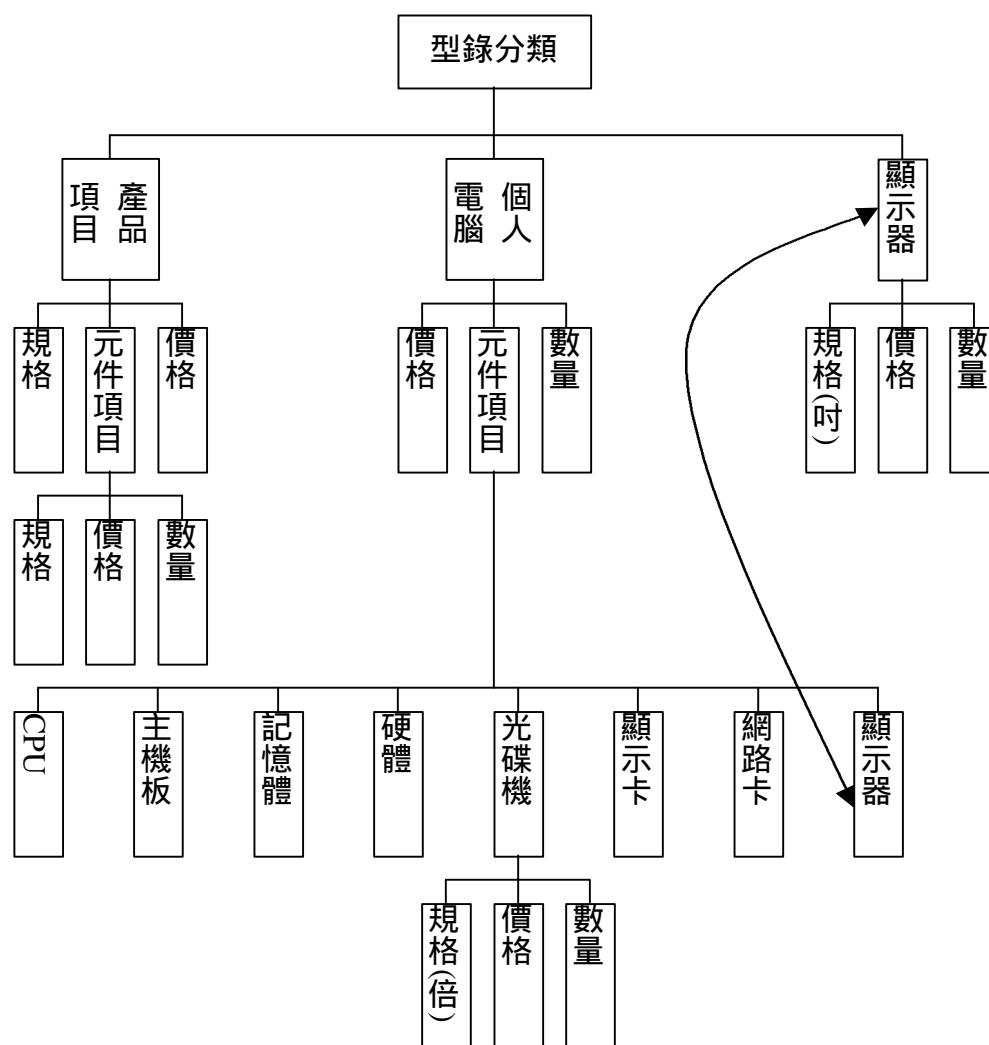


圖 7.2 PC 型錄結構圖

PC 型錄是本系統用戶所期望得到的即時 PC 交易資訊,這裡的交易資訊包含可供貨數量及單項價格。同樣的,本系統需要用戶輸入對該所需商品的數量及預期價格,以作為型錄代理的搜尋參數。但在此之前,參與電子交易市集的廠商必須輸入供應品之資訊至本系統的「供應商資料庫」。在成為電子交易市集會員之前,PC 供應商及 PC 零售商必須提供其 Web 型錄服務的 HTTP URL,以及此 URL 相關的參數名稱及參數值對應市集商品,在會員申請時作註冊並容許在日後進行維護。HTTP URL 是 Web 用戶端向 Web 伺服器端程式提供請求資訊的規則,茲以一家專業代理 CPU 的電腦供應商為例,假設這家名為「供應商 A」的供應商,本身提供 Web 型錄服務,供顧客上網查詢商品現有數量及價格。這個供應商 A 可能提供的型錄服務介面,如同圖 7.3 所示。供應商 A 的 Web 型錄服務的伺服器網址為"localhost",其利用微軟的 ASP 技術來實作 Web 伺服器端程式,以 XML 文件(描述資料內容)及 XSL 文件(描述資料展示方式)的搭配,來達成型錄提供介面。這個 ASP 程式(xmlcatalog.asp)透過商品代號參數(id),參數值決定查詢的標的物,例如欲查詢所有商品則參數值為"all"。依據 IETF RFC 2616 所定義的 HTTP1.1 規範[9],HTTP URL 的定義語法為:

$$\text{http_URL} = \text{"http:"} \text{ " // " } \text{ host [":" port] [abs_path ["?" query]] }$$

其中 host 代表 Web 服務的伺服器網址, port 代表 Web 伺服器的埠號, Web 伺服器的預設埠號為"80", abs_path 代表 Web 伺服器定義的絕對目錄路徑及服務程式的檔案名稱集合, query 是參數名稱及參數值的集合。根據上述 HTTP URL 的定義,則一個顧客欲查詢供應商 A 提供的 Web 型錄服務,此時在 Web 瀏覽器的網址欄輸入"http://localhost/xmlcatalog.asp",即正式向供應商 A 的 Web 型錄服務提出 HTTP 請求,建立顧客的客戶端瀏覽器與供應商 A 的 Web 伺服器端型錄服務程式,彼此互通 HTTP 訊息的連結。這時,透過供應商 A 提供的型錄服務 Web 介面,顧客選擇查詢所有的商品,則其結果同圖 7.3 所示。



圖 7.3 供應商 A 電子型錄介面(I)

透過瀏覽器的產品下拉式選單介面，假設使用者選擇 Pentium III 800 之 CPU，則其畫面如圖 7.4 所示。其中代表商品代號的參數 id，從原先的”all”值變成代表 Pentium III 800 的”1018”編號。圖 7.4 為瀏覽器可見的畫面，而其原始檔，是 XML 配合 XSL 的內容。本例的 XML 內容以表 7.3 加以表示之。在以 Web 為基礎的電子型錄應用中，絕大部數的 Web 應用程式，不論其為 ASP、PHP、JSP 或是 Servlet，其程式執行所需要的參數皆在 HTTP 協定的 URL 中，問號字元後面緊接下來的表示式，例如本例:”id=1018”。所以運用最簡易的觀念可知，供應商如能提供電子交易市集的型錄搜尋代理人系統，關於參數名稱及參數值代表的意義，必能讓型錄搜尋代理人可以透過 HTTP 協定採集供應商的型錄內容。

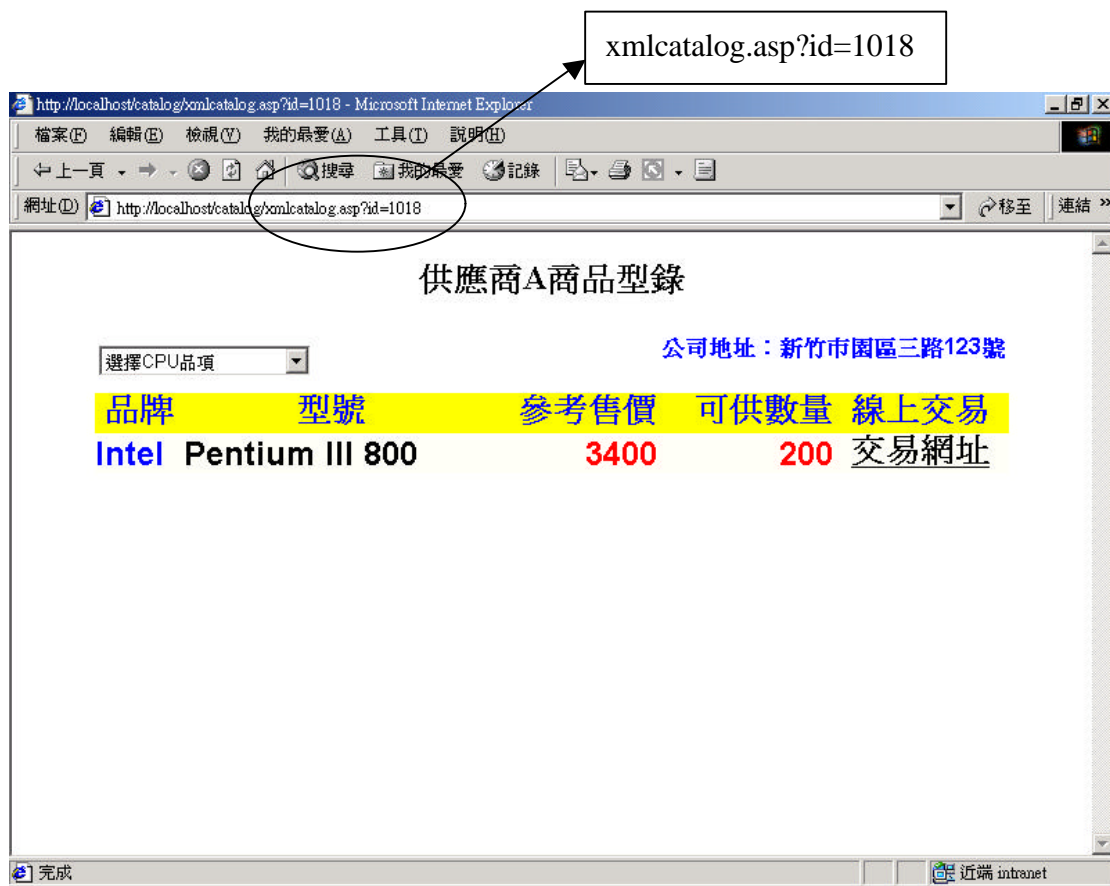


圖 7.4 供應商 A 電子型錄介面(II)

供應商 A 型錄服務的 URL 及商品型錄相關的參數變數名稱，以及參數值對應的商品等資訊，透過電子交易市集中提供的供應商資料維護介面進行維護，並儲存在型錄搜尋代理人中的供應商資料庫。以 HTML 所編寫的供應商資料維護介面，如圖 7.5 所示。供應商需要負責公司基本資料 行動代理人系統位址(ATP)、型錄服務網址(HTTP)、商品參數名稱，商品代號(參數值)以其對應的商品項目。其中行動代理人系統位址，是非同步模式型錄搜尋代理人採集資料的位址，供應商因為可以不提供非同步型錄服務，所以此欄位可以不提供。

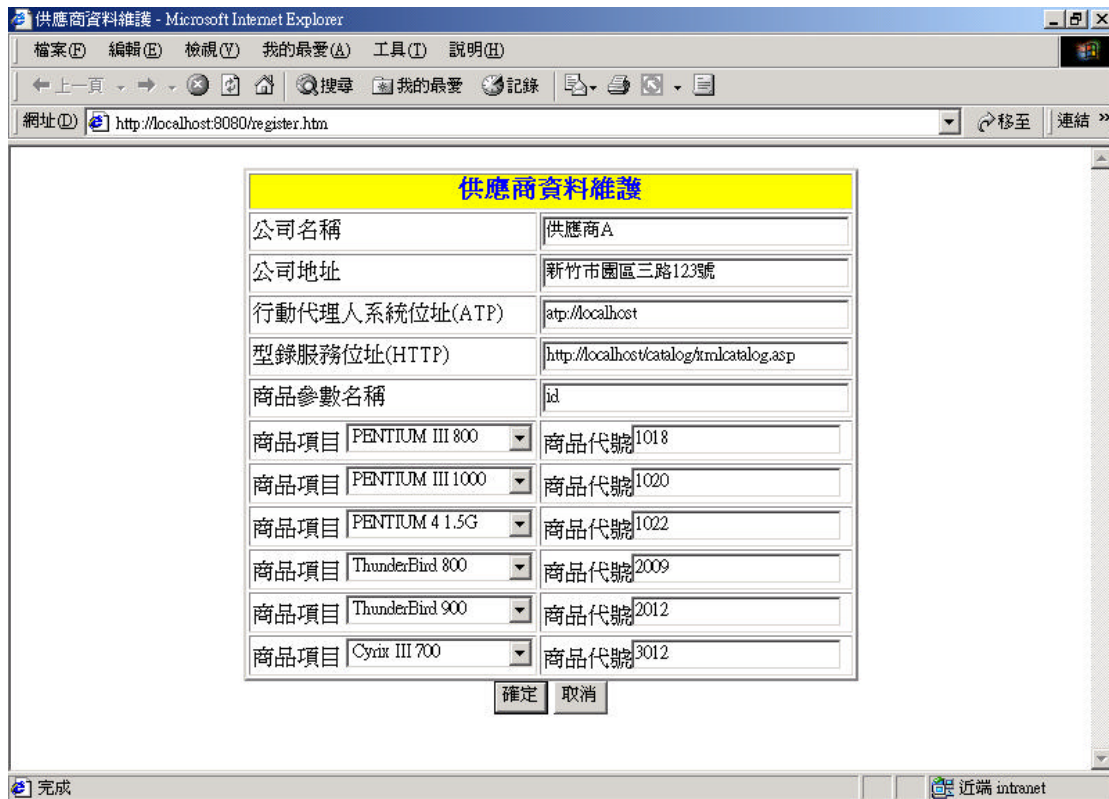


圖 7.5 供應商資料維護介面

供應商資料庫是本系統蒐集供應商型錄之依據。在同步回應模式中，型錄蒐集程式需要供應商資料庫提供的供應商型錄服務之網址，及相關之參數名稱及參數值(商品代號)，以便型錄蒐集程式利用 HTTP 協定向該供應商型錄服務要求商品型錄資訊。在非同步模式中，市集平台中的行動代理人系統亦需要供應商資料庫提供之 ATP 協定位址，才能派遣行動代理人至遠端的供應商行動代理人系統。所以供應商之資料維護，屬於本系統之前置程序，是執行本系統必備的前置過程。

接下來，顧客(PC 零售商或消費者)利用本系統提供之型錄需求規格介面，輸入商品類別、型號(規格)、需求數量、單位預算，成為本系統執行之參數值，此介面如圖 7.6 所示。

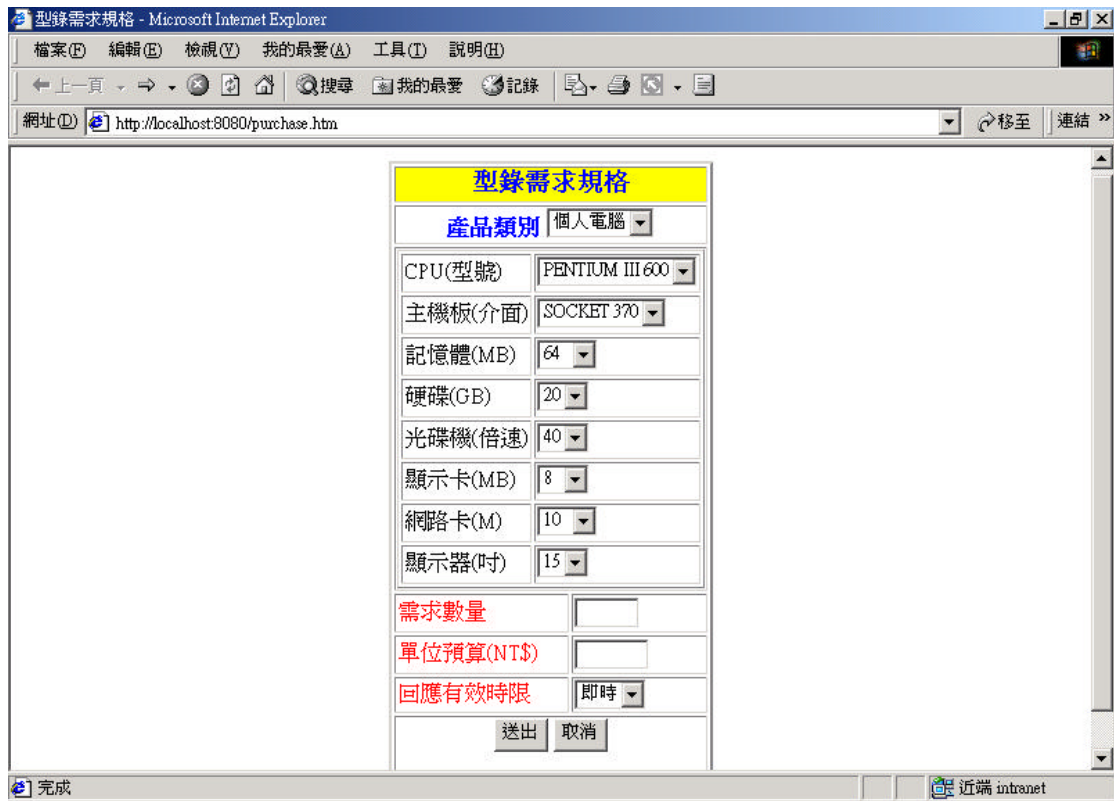


圖 7.6 型錄需求規格介面

由於本系統遵循參與者彈性使用的原則，所以不要求供應商提供的型錄資訊皆必需依據本系統所使用的 RosettaNet PIP3A2 的規格。不同的供應商提供的型錄資訊文件結構可以相異。為求表達本系統的運作流程，茲假設一個情境。

情境描述

一家 PC 零售商，需求 60 個 PENTIUM III 800 CPU，預算單價為 3500 元。並且希望商品型錄資訊在一週內回應。

利用型錄需求規格介面，用戶可以點選產品項目為 CPU，再進一步點選型號為 PENTIUM III 800，然後在需求數量欄填寫 60，在單位預算欄填寫 8000，最後在回應有效時限填入一週，如圖 7.7 所示。

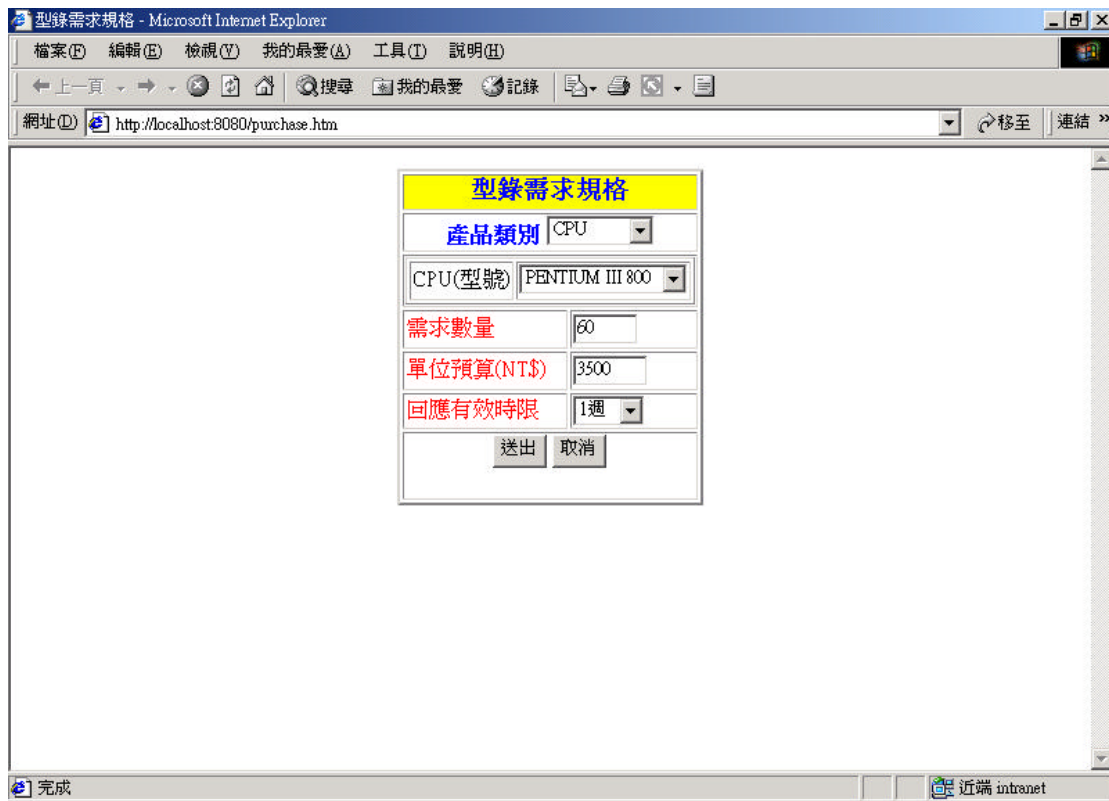


圖 7.7 需求條件輸入畫面

本系統之 Servlet API 接受到用戶在型錄需求介面所輸入的條件之後，開始查詢供應此商品之供應商，將其型錄服務網址傳至同步回應模式的型錄蒐集程式，以及非同步回應模式的行動代理人管理系統。首先，談及同步回應模式。不論用戶輸入的回應有效時限為何，同步回應模式皆會被啟動。同步回應模式的型錄蒐集程式至各個供應商型錄服務平台，利用 HTTP 協定得到的型錄資訊各不相同。假設經過 Servlet API 搜尋供應商資料庫，求得供應商 A、供應商 B、供應商 C 提供 Pentium III 800 CPU。蒐集程式所得到的型錄資訊將以 XML 描述，如表 7.3、表 7.4、表 7.5，分別代表供應商 A、供應商 B、供應商 C 所提供的 Pentium III 800 CPU 之 XML 商品型錄內容。

```

<?xml version="1.0" encoding="Big5" standalone="yes"?>
<?xml-stylesheet href="http://localhost/catalog/SupplierA.xsl"
type="text/xsl" ?>
<catalog>
<product class="CPU">
<brand>Intel</brand>
<type>Pentium III 800</type>
<url>www.timecomputer.com.tw/trade.asp?pd=1018</url>
<condition>
<price>3400</price>
<quantity>100</quantity>
</condition>
</product>
<company>
<name>供應商 A</name>
<address>新竹市園區三路 123 號</address>
</company>
</catalog>

```

表 7.3 供應商 A 的 XML 商品型錄

```

<?xml version="1.0" encoding="Big5" standalone="yes"?>
<ecatalog>
<enterprise>
<enterprise_name>供應商 B</enterprise_name>
<trade_url>www.nctu.com.tw/etrade/catalog?pn=123
</trade_url>
<enterprise_address>新竹市大學路 1001 號
</enterprise_address>
</enterprise>
<product_class>CPU</product_class>
<product>Pentium III 800</product>
<unitprice>3300</unitprice>
<provide_quantity>50</provide_quantity>
</ecatalog>

```

表 7.4 供應商 B 的 XML 商品型錄

```

<?xml version="1.0" encoding="Big5" standalone="yes"?>
<supplier>
<company>供應商 C</company>
<onlineurl>www.superpc.com/trade/trade.jsp?id=pc
</onlineurl>
<physicaladdress>新竹市研新二路 12 號
</physicaladdress>
<product>CPU</product>
<model>Pentium III 800</model>
<price>3700</price>
<availability>200</availability>
</supplier>

```

表 7.5 供應商 C 的 XML 商品型錄

供應商提供的商品型錄資訊是依循各自的標準所建立，它有可能是依循產業的 XML 商務標準，例如：cXML、eCO 及 OBI 等。亦可能是由廠商自行建立的 XML 標準規範，在這裡假設是後者情況(表 7.3 至表 7.5)。由於本系統依據的 XML 商務標準是 RosettaNet PIP3A2，它是本系統回應資訊的標準規範。RosettaNet PIP3A2 包含二個訊息標準，一個是價格及可供數量請求訊息標準(Price And Availability Request Message Guide line)，一個是價格及可供數量回應訊息標準(Price And Availability Response Message Guide line)，分別有其各自的資料型態定義(DTD)，如附表所示。顧名思義，後者是根據前者的內容所作的回應資訊。

在型錄蒐集程式中，RosettaNet PIP3A2 中的價格及可供數量回應訊息標準，是我們期望將供應商提供之商品型錄轉換成的市集標準規範。這個標準的前段是描述商品的價格數量資訊，以<ProductPriceAndAvailability>作為文件的元素標示。轉換後的供應商商品型錄，將如表 7.6 至表 7.8 所示。表 7.6 至表 7.8 所示的 RosettaNet PIP3A2 標準商品型錄分別為表 7.3 至表 7.5 所示的各供應商之商品型錄，將之轉換成 RosettaNet PIP3A2 標準後的商品型錄結果。

型錄蒐集程式的前段程序是至各供應商型錄服務平台，下載以 XML 描述的商品型錄。後端程序則是將所有蒐集的商品型錄，轉換成一致的標準，即 RosettaNet PIP3A2 所明定的規範。這個轉換的工程，交由 IBM XML For Java 中其中一個 API 加以完成，這個 API 為 LMX(Language for Mapping XML)。這裡型錄蒐集程式呼叫這個介面程式，利用它作為 XML to XML 文件轉換的機制，來達成我們將供應商之商品型錄，轉換成市集規格的商品型錄。LMX 以 DOM 標準出發，它是將一個 XML 文件先行轉換成一個標準的 DOM tree。關於 DOM 之相關知識，請參考第四章所述，在此不作贅述。然後再將 DOM tree 的結構，依據一個自訂的轉換規則進行轉換，然後再將新的 DOM tree 結構輸出至新的 XML 文件。例如：型錄蒐集程式下載供應商 A 的商品型錄，下載後呼叫 LMX API 並且載入供應商 A 的型錄轉換規則(toRosettaNet3A2_SupplierA.lmx，如表 7.9 所示) 以及供應商 A 的商品型錄(表 7.3)。然後 LMX API 協助型錄蒐集程式將供應商 A 的商品型錄，轉換成市集所規範的 RosettaNet PIP3A2 的前段部分，如表 7.6 所示。整個運作過程如圖 7.8 所示。

LMX 的轉換規則摘要：

<lmx:lhs>及</lmx:lhs>二個標示中為 XML 來源文件內容，<lmx:rhs>及</lmx:rhs>二個標示中為 XML 目的文件內容。\$var;會分別出現在上述二個標示內容中，代表標示內之變數資料 var 轉換的來源及目的位置。例如：供應商 A 轉換規則中描述<lmx:lhs><price>\$price;</price></lmx:lhs>，以及<lmx:rhs>

<MonetaryAmount>\$price;</MonetaryAmount></lmx:rhs>，將使得供應商 A 商品型錄中的<price>3400</price>經過型錄蒐集程式呼叫執行 LMX API，其結果將變成<MonetaryAmount>3400</MonetaryAmount>。

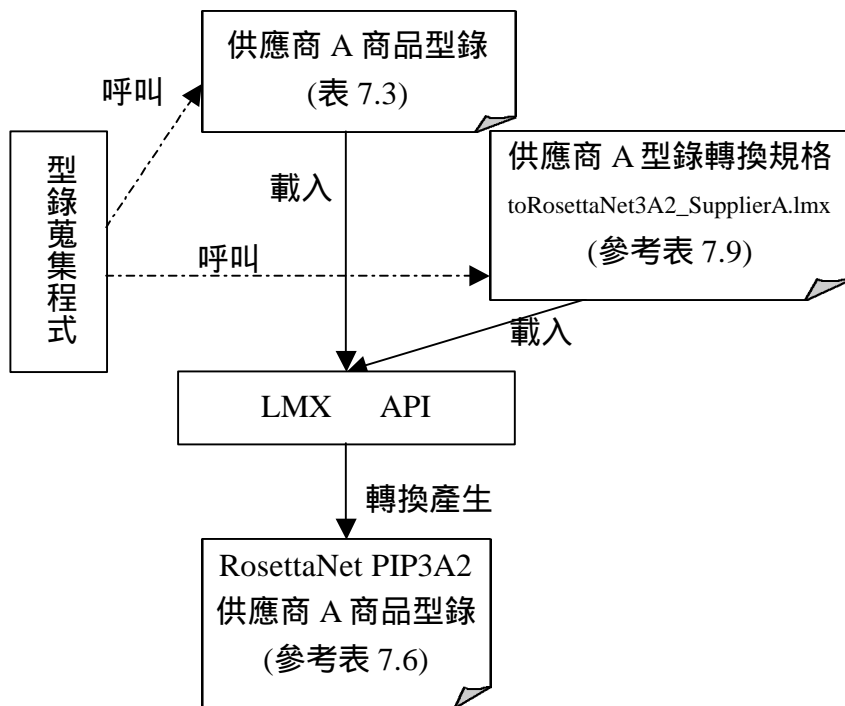


圖 7.8 型錄蒐集程式運作程序

```

<ProductPriceAndAvailability>
<ProductLineItem>
<ProductQuantity>200</ProductQuantity>
<productUnit>
<ProductPackageDescription>
<unitPrice>
<FinancialAmount>
<GlobalCurrencyCode>TWD</GlobalCurrencyCode>
<MonetaryAmount>3400</MonetaryAmount>
</FinancialAmount>
</unitPrice>
<ProductIdentification>
<PartnerProductIdentification>
<GlobalPartnerClassificationCode>Distributor</GlobalPartnerClassificationCode>
<ProprietaryProductIdentifier>Pentium III 800
</ProprietaryProductIdentifier>
</PartnerProductIdentification>
</ProductIdentification>
</ProductPackageDescription>
</productUnit>
</ProductLineItem>
<WarehouseInformationResource>
<warehouseAddress>
<PhysicalAddress>
<addressLine1>供應商 A</addressLine1>
<addressLine2>新竹市園區三路 123 號</addressLine2>
<addressLine3>www.timecomputer.com.tw/trade.asp?pd=1018</addressLine3>
</PhysicalAddress>
</warehouseAddress>
</WarehouseInformationResource>
</ProductPriceAndAvailability>

```

表 7.6 RosettaNet PIP3A2 標準之供應商 A 的 XML 商品型錄

```

<ProductPriceAndAvailability>
<ProductLineItem>
<ProductQuantity>50</ProductQuantity>
<productUnit>
<ProductPackageDescription>
<unitPrice>
<FinancialAmount>
<GlobalCurrencyCode>TWD</GlobalCurrencyCode>
<MonetaryAmount>3300</MonetaryAmount>
</FinancialAmount>
</unitPrice>
<ProductIdentification>
<PartnerProductIdentification>
<GlobalPartnerClassificationCode>Distributor</GlobalPartnerClassificationCode>
<ProprietaryProductIdentifier>Pentium III 800
</ProprietaryProductIdentifier>
</PartnerProductIdentification>
</ProductIdentification>
</ProductPackageDescription>
</productUnit>
</ProductLineItem>
<WarehouseInformationResource>
<warehouseAddress>
<PhysicalAddress>
<addressLine1>供應商 B</addressLine1>
<addressLine2>新竹市大學路 1001 號</addressLine2>
<addressLine3>www.nctu.com.tw/etrade/catalog?pn=123</addressLine3>
</PhysicalAddress>
</warehouseAddress>
</WarehouseInformationResource>
</ProductPriceAndAvailability>

```

表 7.7 RosettaNet PIP3A2 標準之供應商 B 的 XML 商品型錄


```

<ProductPriceAndAvailability>
<ProductLineItem>
<ProductQuantity>200</ProductQuantity>
<productUnit>
<ProductPackageDescription>
<unitPrice>
<FinancialAmount>
<GlobalCurrencyCode>TWD</GlobalCurrencyCode>
<MonetaryAmount>3700</MonetaryAmount>
</FinancialAmount>
</unitPrice>
<ProductIdentification>
<PartnerProductIdentification>
<GlobalPartnerClassificationCode>Distributor</GlobalPartnerClassificationCode>
<ProprietaryProductIdentifier>Pentium III 800
</ProprietaryProductIdentifier>
</PartnerProductIdentification>
</ProductIdentification>
</ProductPackageDescription>
</productUnit>
</ProductLineItem>
<WarehouseInformationResource>
<warehouseAddress>
<PhysicalAddress>
<addressLine1>供應商 C</addressLine1>
<addressLine2>新竹市研新二路 12 號</addressLine2>
<addressLine3>www.superpc.com/trade/trade.jsp?id=pc14</addressLine3>
</PhysicalAddress>
</warehouseAddress>
</WarehouseInformationResource>
</ProductPriceAndAvailability>

```

表 7.8 RosettaNet PIP3A2 標準之供應商 C 的 XML 商品型錄

```

<?xml version="1.0" encoding="Big5"?>
<lmx:rules xmlns:lmx="HTTP://www.ibm.com/xml/lmx/">
<lmx:pattern>
<lmx:lhs>
<catalog><product class="$productname;">
<brand>$brand;</brand><type>$type;</type>
<condition><price>$price;</price>
<quantity>$quantity;</quantity>
<url>$supplier_url;</url></condition>
</product>
<company><name>$supplier_name;</name>
<address>$supplier_address;</address></company>
</catalog>
</lmx:lhs>
<lmx:rhs>
<ProductPriceAndAvailability>
<ProductLineItem><ProductQuantity>$quantity;</ProductQuantity>
<productUnit><ProductPackageDescription><unitPrice>
<FinancialAmount><GlobalCurrencyCode>TWD</GlobalCurrencyCode>
<MonetaryAmount>$price;</MonetaryAmount>
</FinancialAmount></unitPrice>
<ProductIdentification>
<PartnerProductIdentification>
<GlobalPartnerClassificationCode>Distributor</GlobalPartnerClassificationCode>
<ProprietaryProductIdentifier>$type;</ProprietaryProductIdentifier>
</PartnerProductIdentification></ProductIdentification>
</ProductPackageDescription></productUnit>
</ProductLineItem><WarehouseInformationResource>
<warehouseAddress><PhysicalAddress>
<addressLine1>$supplier_name;</addressLine1>
<addressLine2>$supplier_address;</addressLine2>
<addressLine3>$supplier_url;</addressLine3>
</PhysicalAddress></warehouseAddress></WarehouseInformationResource>
</ProductPriceAndAvailability>
</lmx:rhs></lmx:pattern></lmx:rules>

```

表 7.9 供應商 A 型錄轉換規格(toRosettaNet3A2_SupplierA.lmx)

接下來，上述三個完成格式轉換的供應商型錄(表 7.4、表 7.5、表 7.6)，送至型錄代理回應程式進行進一步處理。型錄代理回應程式，主要工作是將這三個型錄根據型錄代理文件規則，進行條件過濾及彙整。條件過濾是依據，需求數 60 單位和預算單價 8000 的 Intel Pentium III 800。將符合此條件的供應商提供之型錄，彙整在回應文件之中。在這個案例中，提供 Intel Pentium III 800 的供應商所提供的單價和數量分別為：供應商 A 提供單價 3400 元，數量 100 個；供應商 B 提供單價 3300 元，數量 50 個；供應商 C 提供單價 3700 元，數量 200 個。經過條件過濾之後，供應商 C 價格過高，供應商 B 數量過少，所以只有供應商 A 符合顧客期待。整個型錄代理回應程式的運作程序，如圖 7.9 所示。型錄代理文件規則，如表 7.10 所示。本程式最後產生的供應商商品型錄的文件內容，如表 7.11 所示。表 7.10 中的 `<embed href="urn:javabean:supplier1#" />` 是符合條件之供應商型錄，將來存在型錄代理回應文件中的所在位置。

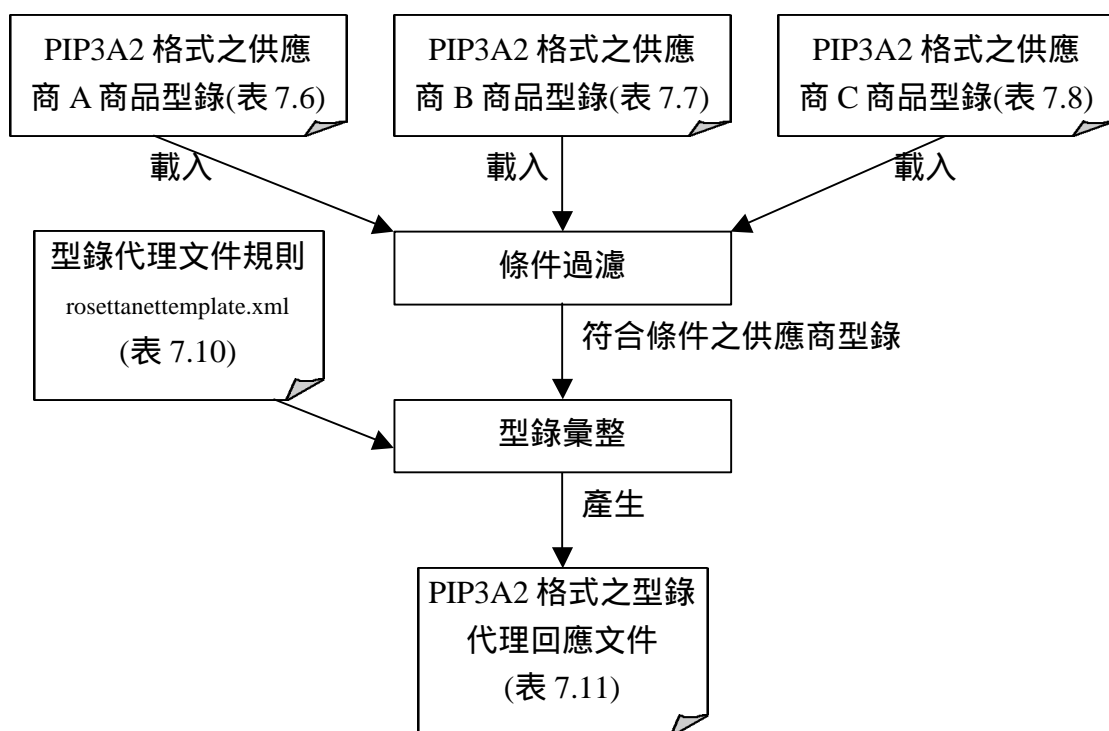


圖 7.9 型錄代理回應程式運作程序

```

<?xml version="1.0" encoding="Big5"?>
<?xml-stylesheet href="HTTP://localhost:8080/RosettaNet.xml" type="text/xml" ?>
<Pip3A2PriceAndAvailabilityResponse>
<embed href="urn:javabean:supplier1#"/>
<fromRole>
<PartnerRoleDescription>
<GlobalPartnerRoleClassificationCode>Vendor</GlobalPartnerRoleClassificationCode>
<PartnerDescription>
<BusinessDescription>
<GlobalBusinessIdentifier>123456789</GlobalBusinessIdentifier>
<GlobalSupplyChainCode>Information Technology</GlobalSupplyChainCode>
</BusinessDescription></PartnerDescription>
<ContactInformation><contactName>wake lin</contactName>
<telephoneNumber><CommunicationsNumber>0226221111</CommunicationsNumber>
<EmailAddress>maddux.lin@msa.hinet.net</EmailAddress>
</telephoneNumber></ContactInformation></PartnerRoleDescription></fromRole>
<toRole><PartnerRoleDescription>
<GlobalPartnerRoleClassificationCode>Customer</GlobalPartnerRoleClassificationCode>
<PartnerDescription><BusinessDescription>
<GlobalBusinessIdentifier>987654321</GlobalBusinessIdentifier>
<GlobalSupplyChainCode>Information Technology</GlobalSupplyChainCode>
</BusinessDescription></PartnerDescription></PartnerRoleDescription>
</toRole><thisDocumentGenerationDateTime>
<DateTimeStamp>20010310T103000.000Z</DateTimeStamp>
</thisDocumentGenerationDateTime><thisDocumentIdentifier>
<ProprietaryDocumentIdentifier>20010310001</ProprietaryDocumentIdentifier>
</thisDocumentIdentifier><requestingDocumentIdentifier>
<ProprietaryDocumentIdentifier>20010310001</ProprietaryDocumentIdentifier>
</requestingDocumentIdentifier>
<GlobalDocumentFunctionCode>Response</GlobalDocumentFunctionCode>
<requestingDocumentDateTime>
<DateTimeStamp>20010310T103000.000Z</DateTimeStamp>
</requestingDocumentDateTime>
</Pip3A2PriceAndAvailabilityResponse>

```

表 7.10 型錄代理文件規則(rosettanettemplate.xml)

```

<?xml version="1.0" encoding="Big5"?>
<?xml-stylesheet href="HTTP://localhost:8080/RosettaNet.xml" type="text/xml" ?>
<Pip3A2PriceAndAvailabilityResponse>
<ProductPriceAndAvailability>
<ProductLineItem>
<ProductQuantity>100</ProductQuantity>
<productUnit>
<ProductPackageDescription>
<unitPrice>
<FinancialAmount><GlobalCurrencyCode>TWD</GlobalCurrencyCode>
<MonetaryAmount>3400</MonetaryAmount></FinancialAmount>
</unitPrice>
<ProductIdentification>
<PartnerProductIdentification>
<GlobalPartnerClassificationCode>Distributor</GlobalPartnerClassificationCode>
<ProprietaryProductIdentifier>Pentium III 800
</ProprietaryProductIdentifier></PartnerProductIdentification>
</ProductIdentification></ProductPackageDescription>
</productUnit></ProductLineItem>
<WarehouseInformationResource>
<warehouseAddress>
<PhysicalAddress>
<addressLine1>供應商 A</addressLine1>
<addressLine2>新竹市園區三路 123 號</addressLine2>
<addressLine3>www.timecomputer.com.tw/trade.asp?pd=1018</addressLine3>
</PhysicalAddress></warehouseAddress>
</WarehouseInformationResource></ProductPriceAndAvailability>
<fromRole>
.....與表 6.8 相同
</Pip3A2PriceAndAvailabilityResponse>

```

表 7.11 PIP3A2 格式之型錄代理回應文件

在表 7.10 及表 7.11 中的第二段，代表本文件的排版規則，內容為
<?xml-stylesheet href="HTTP://localhost:8080/RosettaNet.xml" type="text/xml" ?>，
RosettaNet.xml 是一個 XSL 文件，用來描述本 XML 文件在瀏覽器上的 HTML 排

版規則，其內容以表 7.12 加以表示。

```
<xsl:stylesheet xmlns:xsl="HTTP://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML><BODY><STYLE>td { font: 9pt Arial; color: Black; text-align: left;}</STYLE>
<TABLE BORDER="1" CELLPADDING="6">
<TR><TH>產品名稱</TH><TH>單價</TH><TH>數量</TH>
<TH>供應商</TH><TH>交易網址</TH><TH>地址</TH></TR>
<xsl:for-each select="Pip3A2PriceAndAvailabilityResponse/ProductPriceAndAvailability">
<TR><xsl:apply-templates /></TR>
</xsl:for-each></TABLE></BODY></HTML></xsl:template>
<xsl:template match="ProductLineItem">
<TD STYLE="color:BLUE;">
<xsl:value-of
select="productUnit/ProductPackageDescription/ProductIdentification/PartnerProductIdentificatio
n/ProprietaryProductIdentifier"/></TD>
<TD STYLE="color:RED;">
<xsl:value-of
select="productUnit/ProductPackageDescription/unitPrice/FinancialAmount/MonetaryAmount"/>
</TD>
<TD STYLE="color:RED;"><xsl:value-of select="ProductQuantity"/></TD>
</xsl:template>
<xsl:template match="WarehouseInformationResource">
<TD STYLE="color:BLUE;">
<xsl:value-of select="warehouseAddress/PhysicalAddress/addressLine1"/></TD>
<TD STYLE="color:BLUE;">
<xsl:element name="A">
<xsl:attribute name="href">HTTP://
<xsl:value-of select="warehouseAddress/PhysicalAddress/addressLine3"/>
</xsl:attribute><xsl:value-of select="warehouseAddress/PhysicalAddress/addressLine3"/>
</xsl:element>
</TD> <TD STYLE="color:BLUE;">
<xsl:value-of select="warehouseAddress/PhysicalAddress/addressLine2"/>
</TD></xsl:template></xsl:stylesheet>
```

表 7.12 型錄代理回應文件排版樣本(RosettaNet.xsl)

最後，型錄代理回應文件(表 7.11)配合其排版樣本(表 7.12)，在用戶端瀏覽器展示的結果，將如同圖 7.10 之畫面所示。用戶可以看到其所設定條件回應的商品型錄即時資訊，這些資訊可以作為用戶決定交易對象的參考依據。並且可以點選該供應商提供的交易網址直接與該供應商進行網上交易，網上交易的方法依不同供應商而相異。本系統旨在避免與供應商爭利，所以不提供交易機制。

產品名稱	單價	數量	供應商	交易網址	地址
Pentium III 800	3400	200	供應商A	交易	新竹市園區三路123號

圖 7.10 本系統回應用戶之商品型錄畫面

在談述完本系統同步回應模式的細部實作流程後，接下來開始探討非同步回應模式的實作流程，亦即行動代理人架構的模式之細部作業。行動代理人架構需要各部主機安裝行動代理人系統，在這裡選用的行動代理人系統，稱作 IBM Aglets。運作此非同步回應模式，需要在電子交易市集及供應商平台上安裝 IBM Aglets 行動代理人系統，在顧客端或稱作本系統的用戶端不需要安裝 IBM

Aglets。IBM Aglets 是以端對端為基礎的分算式運算平台，在本系統中，我們期望透過電子交易市集來派遣一個查詢供應商平台的商品型錄搜尋代理人，使得電子交易市集與供應商的行動代理人系統，進行端對端的訊息交換。為強調用戶端最小安裝為原則，減低用戶使用電子交易市集系統的進入學習障礙，所以本系統利用 Web 技術與行動代理人系統整合，以達成非同步回應用戶商品型錄之目標。整合 Web 技術的行動代理人系統，其操作流程請參考圖 6.7 所示。

在先前的應用情境中，我們假設顧客提出的需要是 60 個單價 8000 的 Intel Pentium III 800 的 CPU，且回應有效時限是一週並非只要即時。所以不但先前的同步回應模式需要啟動運作，這裡提到的非同步回應模式亦需要被本系統執行。所以在行動代理人被派遣至遠端的供應商行動代理人系統之後，即開始每隔一個小時對供應商的 Web 伺服器型錄服務，進行 HTTP 請求並且接受商品型錄的 XML 回應。如果行動代理人查核商品型錄中的單價或數量，符合顧客限定的條件，則行動代理人將被從供應商平台中移除，並將最新的型錄資訊傳回至電子交易市集平台的行動代理人系統。之後行動代理人啟動通知程式，以 SMTP 協定傳送電子郵件至顧客所設定的電子郵件信箱，其所在的郵件伺服器。顧客端的用戶利用其電子郵件系統，利用 POP3 協定向郵件伺服器下載電子郵件，即能接受電子交易市集商品型錄的通知訊息，然後直接點選電子郵件中記載的供應商交易網址，使顧客可以與供應商藉此進行雙邊之交易。

上述二種模式的交互運作，使得本系統可以提供顧客同步(即時)回應的商品型錄，或是非同步(未來)回應的商品型錄。這個系統最大的優點在於，提供分散式資訊蒐整的架構，使得供應商商品型錄不是交易供應商自行透過電子交易市集的 Web 系統，進行線上型錄維護。而是將供應商的型錄資料庫，視為電子交易市集上型錄服務資料庫的一部分，雖然這個資料庫並非存在電子交易市集平台上，但卻能更正確即時地將商品型錄資訊回應給顧客，所以這裡將電子交易市

集提供的型錄服務稱作型錄代理，亦即電子交易市集代理顧客的身分，至供應商平台搜尋對顧客有價的即時型錄資訊。另一方面，本系統提供的非同步回應模式，不同於大部分的三層式資訊系統，是將型錄蒐集程式在本地端(電子交易市集平台)執行。如此的話，則型錄蒐集程式必須透過網路，下載遠端(供應商平台)的型錄資訊才能進行型錄蒐集的目的，可能因此而造成網路資源的過度使用。相反的，本系統利用行動代理人架構，使得型錄蒐集程式是在遠端(供應商平台)執行。如此的話，則型錄蒐集程式只會在蒐集結果符合所需時才會將型錄資訊傳回，因此不會有網路負荷過大的問題。綜合這二點，結論為效率和品質上的改良是本系統最重要的優點。另外，利用 Web 技術提升行動代理人系統的接受度，亦即用戶端可以最小安裝。以及行動代理人所擷取之資源為遠端之 Web 應用程式，而非自行建立檔案或資料庫系統。更使得行動代理人系統，能符合資訊系統運作之彈性、使用者親和性的要求，以提升本系統之市場接受度。

唯本系統雖然具備上述的優點，但是在功能上尚有二個待改進的地方，可以作為未來改版精進的參考。本系統接受用戶在註冊階段，利用供應商會員註冊介面，以使其銷售的商品類別可以與市集中型錄服務的商品類別作一對應，以作為市集中的型錄搜尋代理人向供應商平台蒐集型錄資訊的依據。但是這個介面面臨一個挑戰，即如果一家供應商銷售的商品類別非常多且複雜，那麼透過本系統提供的標準介面，不易作大量商品對應資料的維護，這點可以從改良使用者介面加以改進。或是未來國際或國內對資訊產業的商品制定一致的編碼序號，那麼各個供應商的型錄服務與市集中的型錄搜尋代理人，就不難建立關聯。另外，尚需改進的地方是文件轉換規則的自動化。本型錄搜尋代理人系統，必須將所有提供不同規則的型錄，依據文件轉換規則將該型錄轉譯成 RosettaNet PIP3A2 規格的 XML 文件。這部分目前本系統是仰賴電子交易市集的管理者，透過人工維護的方式來達成，但是如此作法易有人為疏失造成錯誤，以及不易掌握供應商型錄規格變動的缺點。所以未來將文件轉換規則利用程式自動化維護，將可改善這個缺點。

第八章 結論與未來展望

本論文所提出的型錄搜尋代理人架構，整合行動代理人與 XML 在分散式運算及應用程式整合上的優勢而成。由於傳統利用行動代理人建構電子交易市集的方法，需要參與者建構一致的市集平台，使得用戶失去開發系統的主控權因此缺乏彈性。而 XML 建構在三層式的 Web 架構中，雖然既簡單又具彈性，但無法如行動代理人可以讓應用系統具備高效率及高服務品質。因而整合行動代理人與 XML 架構，將可以彌補雙方的缺點，達到相得益彰的效果。

整合架構的可用性必須要實作來驗證，而本論文在最後利用 Java 作為系統開發平台，建構一個型錄搜尋代理人雛型系統。本架構有四個組成要件，稱為 Java 應用程式介面，包含 Aglets、XML for Java、Servlet、JDBC。Aglets 及 XML for Java 提供本系統良好的行動代理人及 XML 的開發環境，讓本系統得以依據架構的理念而實作。再加上 Java 本身提供的 Web 伺服器應用程式開發環境 - Servlet，提供有效的機制使內部 Java 可以與外部的用戶端請求搭配執行。JDBC 提供 Java 應用程式存取資料庫的機制。利用這個實作完成的系統，配合一個 PC 的採購應用情境，以瞭解本論文所提的整合架構的貢獻及優點。在可預期的數年內，電子交易市集將會有更精進的發展，改變全球市場行銷及全球運籌的型態。在 WTO 開放後，電子交易市集的運作將使我國企業的交易夥伴遍及世界各角落。在國內建構電子交易市集的優勢，在於台灣是製造業重鎮，而電子交易市集提供其供應鏈效率。即便現今國內的製造業工廠不斷移往大陸，但是由台灣接單大陸生產的模式，有了電子交易市集更能提供廠商產品的能見度，提高客戶下單的機會。然而，提供型錄服務是電子交易市集的基本機能，有效提供客戶加值的型錄服務，筆者認為在電子交易市集萌芽的現階段，是最重要的議題。所以本論文提出的型錄搜尋代理人架構，即是應用在公眾式的電子交市集中提供其型錄模式架構方案。

雖然電子交易市集真正的未來發展未可預知，無法預判何種模式的電子交易市集會成功。即便如此，專業分析單位肯定電子交易市集的價值，認為它是電子商務發展的一股重要推力。透過本論文所提出的架構，讀者對電子交易市集在型錄搜尋的附加價值，應會持有更樂觀的信心和期待。未來此架構能夠擴充的服務還很多，可以視產業環境的狀況而異。例如在本架構非同步回應模式中，在供應商平台上的代理人程式，除了可以執行現有的型錄蒐集功能外，它可以做到一個反向拍賣的功能。作法是在供應商平台上，建置一個靜態議價代理人，透過與儲存顧客需求價格的行動代理人作溝通，讓供應商有機會調整價格以促成顧客下單的目的。另外，在本架構的實作系統上，目前並沒有安全服務機制。然而，由於系統完全使用 Java 平台實作，而 Java 核心以及 XML for Java 提供的 API，內含安全服務相關的介面，包含數位簽章、Hash 等功能，是未來擴充安全服務的有利條件。在未來，欲使本型錄搜尋代理人架構，可以卓越於其他架構之上，必須改善其便利性以提升其可用性。便利性欠佳之處，是在以 XML 描述的型錄文件格式的轉換規則的處理，目前文件轉換規則僅能透過人為介入方能產生。所以轉換規則的自主化、自動化維護處理，將是未來此架構應行改善之處。

參考文獻

- [1] A. L. Hors, P. L. Hegaret, L. Wood, G. Nicol, J. Robie, "Document Object Model (DOM) Level 2 Core Specification Version 1.0", W3C Recommendation, 13 November 2000.
- [2] Box, Ehnebuske, Kakivaya, Layman, Mendelsohn, Nielsen, Thatte Winer, "Simple Object Access Protocol (SOAP)", W3C Note, 08 May 2000.
- [3] Brain E. Travis, "XML and SOAP Programming for Biztalk Servers", September 2000.
- [4] Carlo Ghezzi, Giovanni Vigna, "Mobile Code Paradigms and Technologies: A Case Study", First International Workshop on Mobile Agents 97 (MA' 97), Berlin, April 7-8, 1997.
- [5] David Manion, "Asia Pacific Supply Chain Survey Executive Summary", CommerceNet Research Report Research Bulletin #99-35, 19 November 1999.
- [6] D.B. Lange and M. Oshima, "Programming and Deploying Java Mobile Agents with Aglets", Addison-Wesley, 1998.
- [7] D.B. Lange and M. Oshima, "Seven good reasons using mobile agent", Communication of the ACM, vol. 42, 1999, pp.88-89.
- [8] D. Kotz et al., "Agent TCL: Targeting the Needs of Mobile Computers", IEEE Internet Computing, vol. 1, no.4, July-Aug. 1997, pp.58-67.
- [9] Fielding, Gettys, Mogul, Frystyk, Masinter, Leach, Lee, "Hypertext Transfer Protocol – HTTP/1.1", IETF 2616, 1999.
- [10] General Magic, "<http://www.genmagic.com/>".
- [11] Geoffrey Lewis, Steven Barber, Ellen Siegel, "Programming with Java IDL", 1998.
- [12] Giacomo Cabri, L. Leonardi, F. Zambonelli, "Coordination Models for Internet

- Application based on Mobile Agents", IEEE Computer Magazine, 1999.
- [13] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, "XML and Java Developing Web Application", January 2000.
- [14] Holger Peine and Torsten Stolpmann, "The Architecture of the Ara Platform for Mobile Agents", First International Workshop, MA '97, Berlin, Germany, April 7-8, 1997 Proceedings.
- [15] Hua L. Lee, Seungjin Whang, "Supply Chain Integration Over the Internet", CommerceNet, 8 January 2001.
- [16] H.S. Nwana, D.T. Ndumu, "Agent Technology", Springer, December 1997, pp.29-41.
- [17] IBM, "Aglets Documentation", January 1999, "<http://www.trl.ibm.co.jp/aglets>".
- [18] Jason Hunter, William Crawford, "Java Servlet Programming", O' Reilly, 1999.
- [19] Mark Norris, Steve West, Kevin Gaughan, "eBusiness Essentials – Technology and Network Requirements for the Electronic Marketplace", John Wiley, 2000.
- [20] Mitsubishi Electric, "Concordia: An Infrastructure for Collaborating Mobile Agents", First International Workshop on Mobile Agents 97 (MA' 97), Berlin, April 7-8, 1997.
- [21] Odyssey, "<http://www.generalmagic.com/technology>".
- [22] Pattie Maes, Robert H. Guttman, and Alexandaros G. Moukas, "Agents That Buy and Sell", Communication of the ACM, vol. 42, 1999, pp.81-89.
- [23] Paulo Jorge Marques, Luis Moura Silva, Joao Gariel Silva, "Security Mechanisms for Using Mobile Agents in Electronic Commerce", Reliable Distributed Systems, 1999. Proceedings of the 18th IEEE Symposium on , 1999. pp. 378 –383.
- [24] Robert J. Glushko, Jay M. Tenenbaum, Bart Meltzer, "An XML framework for Agent-based E-commerce", Communication of the ACM, vol. 42, 1999, pp.106-114.
- [25] RosettaNet, "<http://www.rosettanet.org>".

- [26] Shim, S.S.Y.; Pendyala, V.S.; Sundaram, M.; Gao, J.Z, “Business-to- business e-commerce frameworks”, Computer , Volume: 33 Issue: 10 , Oct. 2000, pp. 40-47.
- [27] Sooho Sohn, Kwan Jong Yoo, ” An architecture of electronic market applying mobile agent technology”, Computers and Communications, 1998. ISCC '98. Proceedings. Third IEEE Symposium on , 1998, pp. 359 –364.
- [28] Steve Terry, “The Development of Online Marketplaces”, CommerceNet Research Report Research Bulletin #99-20, July 1999.
- [29] Ted Pattison, ”Programming Distributed Application with COM and Microsoft Visual Basic 6.0”, Microsoft, 1999.
- [30] Tim Bray, Jean Paoli, C. M. Sperberg, Eve Maler, “ Extensible Markup Language 1.0 (Second Edition) ” , W3C Recommendation, 6 October 2000.
- [31] William R. Cockayne, Michael Zyda, “Mobile Agents”, Manning, 1998, pp.17-19.
- [32] 電子時報 , ”<http://www.digitimes.com.tw>”。
- [33] XML 台灣資訊網 , ”<http://www.xml.org.tw>”。
- [34] Growbal 著 , 2001 年 3 月 1 日 , 官欣怡校閱「跨平台 e 化程式設計-XML/Java/SOAP 整合應用」。
- [35] 蔡桂芳、萬洪壽 , 2000 年 10 月 , 「B2B 虛擬商場完全經營手冊」, 商智文化。
- [36] 經濟部工業局 , 2000 年 12 月 1 日 , 「推動台灣產業網路電子市集計畫」, 經濟部工業局新聞稿。

